



UNIVERSITÀ DI PARMA

Analisi dei Sistemi Informativi

Overview della fase di analisi

- L'obiettivo è di sviluppare una comprensione chiara dei requisiti del nuovo sistema
 - Comprendere il sistema “As-Is”
 - Identificare i miglioramenti
 - Sviluppare il concetto del sistema “To-Be”
- Utilizzare le abilità del pensiero critico per determinare le vere cause dei problemi
- Applicare le conoscenze dei SI e del business per delineare i modi per risolvere i problemi nel nuovo sistema

Analisi dei requisiti

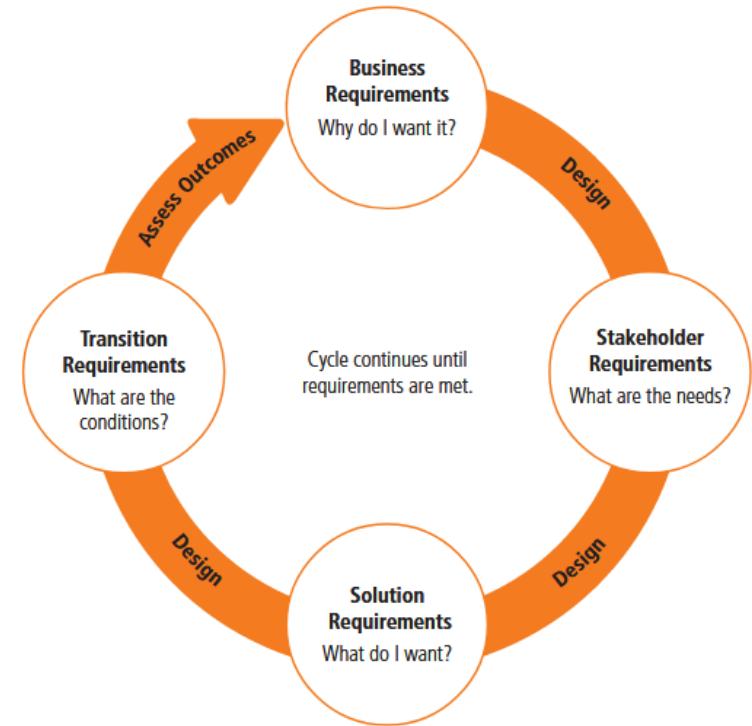
- Anche chiamata **Requirement Engineering**.
- In sostanza è l'analisi business orientata alla **progettazione di soluzioni IT** per la **risoluzione** delle **esigenze** che emergono
- Deve **raccogliere i requisiti** che servono al progetto IT successivo

Requisito? (IEEE 610,12-1.990)

- Una **condizione o capacità necessaria** per una delle parti interessate per **risolvere un problema o raggiungere un obiettivo**;
- Una condizione o capacità che deve essere **soddisfatta o posseduta da un sistema o componente del sistema** per soddisfare **un contratto, standard, specifiche o altri documenti formalmente istituiti**;
- Una **rappresentazione documentata** di una condizione o capacità come ai punti precedenti.

Tipologie di requisiti

- Requisiti di **business**
- Requisiti degli **stakeholder**
- Requisiti della **soluzione**
 - funzionali
 - non funzionali/**quality of service**
- Requisiti di **transizione**



Da "A Guide to the Business Analysis Body of Knowledge - IIBA

Suddivisione dei requisiti

- Cosa necessita al business (requisiti di business);
- Cosa deve fare l'utente (requisiti utente);
- Cosa dovrebbe fare il software (requisiti funzionali);
- Caratteristiche che il sistema dovrebbe avere (requisiti non funzionali);
- Come dovrebbe essere costruito il sistema (requisiti di sistema)
- +Assumption e Constraint (ipotesi e vincoli)

Categorizzazione dei requisiti

- Requisiti **funzionali esplicativi**
- Requisiti **funzionali impliciti**
- Requisiti **non funzionali esplicativi**
- Requisiti **non funzionali impliciti**

Requisiti funzionali

- Un processo** che il sistema dovrebbe eseguire come parte del supporto a un compito utente
- Informazione** che il sistema dovrebbe fornire mentre l'utente esegue un compito
- Specificano il supporto che il sistema fornirà all'utente per soddisfare il proprio compito di lavoro

Ancora sui requisiti funzionali

- Process-oriented
- Information-oriented

Functional Requirement	Description	Examples
Process-oriented	A process the system must perform; a process the system must do	<ul style="list-style-type: none">• The system must allow registered customers to review their own order history for the past 3 years.• The system must check incoming customer orders for inventory availability.• The system should allow students to view a course schedule while registering for classes.
Information-oriented	Information the system must contain	<ul style="list-style-type: none">• The system must retain customer order history for 3 years.• The system must include real-time inventory levels at all warehouses.• The system must include budgeted and actual sales and expense amounts for the current year and 3 previous years.

Requisiti non funzionali

- Proprietà comportamentali che il sistema deve avere
 - Operative – ambiente operativo fisico e tecnico
 - Prestazioni – velocità, capacità e affidabilità di cui necessita il sistema
 - Sicurezza – restrizione d'accesso, protezioni necessarie
 - Politiche e culturali – problematiche che influenzano il sistema finale

Esempi di requisiti non funzionali

Proprietà comportamentali che il sistema deve avere

Nonfunctional Requirement	Description	Examples
Operational	The physical and technical environments in which the system will operate	<ul style="list-style-type: none">The system will run on Android mobile devices.The system should be able to integrate with the existing inventory system.The system should be compatible with any Web browser.Any interaction between the user and the system should not exceed 2 seconds.The system downloads new status parameters within 5 minutes of a change.The system should be available for use 24 hours per day, 365 days per year.The system supports 300 simultaneous users from 9–11 A.M.; 150 simultaneous users at all other times.Only direct managers can see staff personnel records.Technicians can see only their own work assignments.The system includes all available safeguards from viruses, worms, Trojan horses, etc.
Performance	The speed, capacity, and reliability of the system	
Security	Who has authorized access to the system under what circumstances	
Cultural and Political	Cultural and political factors and legal requirements that affect the system	<ul style="list-style-type: none">The system should be able to distinguish between US currency and currency from other nations.Company policy is to buy computers only from Dell.Country managers are permitted to authorize custom user interfaces within their units.Personal information is protected in compliance with the Data Protection Act.

Source: The Atlantic Systems Guild, www.systemsguild.com

Documentare i requisiti

- Rapporto sulla definizione dei requisiti**
 - Documento testuale che elenca i requisiti in forma sintetica
 - Organizzato in raggruppamenti logici
 - Possono essere incluse le priorità
- Uno scopo chiave è definire l'ambito (scope) del progetto**
 - Cosa è incluso
 - Cosa è escluso

Requisito di qualità (BABOK)

- **Atomic**: self-contained and capable of being understood independently of other requirements or designs.
- **Complete**: enough to guide further work and at the appropriate level of detail for work to continue. The level of completeness required differs based on perspective or methodology, as well as the point in the life cycle where the requirement is being examined or represented.
- **Consistent**: aligned with the identified needs of the stakeholders and not conflicting with other requirements.
- **Concise**: contains no extraneous and unnecessary content.
- **Feasible**: reasonable and possible within the agreed-upon risk, schedule, and budget, or considered feasible enough to investigate further through experiments or prototypes.
- **Unambiguous**: the requirement must be clearly stated in such a way to make it clear whether a solution does or does not meet the associated need.
- **Testable**: able to verify that the requirement or design has been fulfilled. Acceptable levels of verifying fulfillment depend on the level of abstraction of the requirement or design.
- **Prioritized**: ranked, grouped, or negotiated in terms of importance and value against all other requirements.
- **Understandable**: represented using common terminology of the audience

Requisito SW di qualità (IEEE Std 830-1998)

IEEE 830-1998 - *IEEE Recommended Practice for Software Requirements Specifications*

- **Corretto** (REQ *Car rental prices shall show all applicable taxes (including 6% state tax).*)
- **Non ambiguo** (REQ *The system shall not accept passwords longer than 15 characters. If the user enters more than 15 characters while choosing the password, an error message shall ask the user to correct it.*)
- **Completo** (REQ1 *A destination country does not need to be displayed for flights within the U.S.* REQ2 *For overseas flights, the system shall display a destination country. What about flights to Canada/Mexico ?*)
- **Consistente** (REQ1 *Payment by PayPal shall be available.* REQ2 *Only credit card payments shall be accepted.*)
- **Ordinato per importanza(priorità) e stabilità**
- **Verificabile** (REQ *The search facility should allow the user to find a reservation based on Last Name, Date, etc.*)
- **Modificabile** (non deve avere eccessive dipendenze)

Fattibile (REQ *The system shall have a natural language interface that will understand commands given in English language*)

Requisiti vs. Progettazione

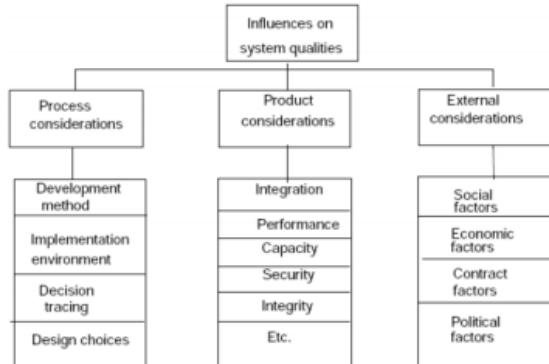
- Requisiti:
 - **Cosa** il sistema deve fare
 - Astrazione



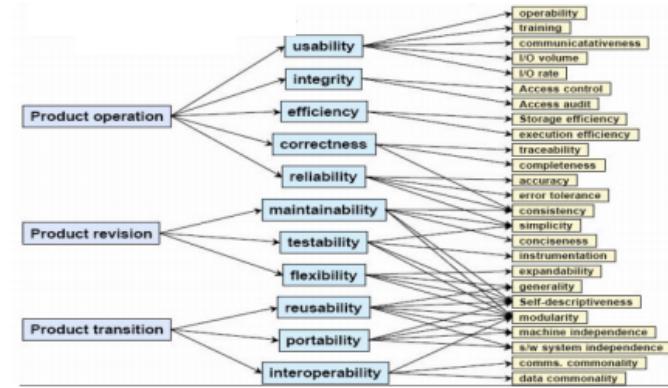
- Progettazione:
 - **Come** il sistema deve farlo
 - Dettagli (anche tecnici)



Classificazioni dei requisiti



Sommerville, 1992



Van Vliet, 2000

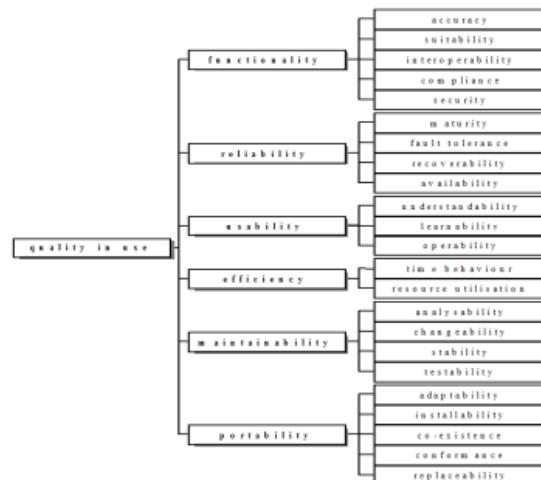
The FURPS+ System for Classifying Requirements

- Functionality
- Usability
- Reliability
- Performance
- Supportability

The "+" in FURPS+ also helps us to remember concerns such as:

- Design requirements
- Implementation requirements
- Interface requirements
- Physical requirements

Unified Process



ISO 9126

Classificazioni dei requisiti

	McCall 1980	Keller 1990	Deutsch 1988	ISO/IEC 9126; 1991	Sommerville 1992	Gilb 2005	Miller 2009
Availability						x	x
Correctness	x	x	x				
Efficiency	x	x	x	x	x	x	x
Expandability		x	x				x
Flexibility	x	x	x			x	x
Integrity	x	x	x			x	x
Interoperability	x	x	x	x	x		x
Maintainability	x	x	x	x		x	x
Portability	x	x	x	x	x		x
Reliability	x	x	x	x	x	x	x
Reusability	x	x	x				x
Safety			x		x		
Security				x			x
Survivability		x	x				x
Testability	x			x			
Usability	x	x	x	x	x	x	x
Verifiability		x	x				x

The Quest for Software Requirements, Roxanne Miller

<https://requirementsquest.com/nonfunctional-requirement-examples/>

L'elicitazione dei requisiti

- Utilizzare ogni interazione con dirigenti e utenti per ricavare interesse, supporto, e entusiasmo per il progetto
- Scegliere i partecipanti con cura
- Fare un uso rispettoso del tempo delle persone

Metodi per l'elicitazione dei requisiti

- Interviste
- JAD – Joint Application Development
- Questionari
- Osservazione
- Analisi di documenti

	Interviews	Joint Application Design	Questionnaires	Document Analysis	Observation
Type of information	As-is, improvements, to-be	As-is, improvements, to-be	As-is, improvements	As-is	As-is
Depth of information	High	High	Medium	Low	Low
Breadth of information	Low	Medium	High	High	Low
Integration of information	Low	High	Low	Low	Low
User involvement	Medium	High	Low	Low	Low
Cost	Medium	Low–Medium	Low	Low	Low–Medium

Interviste

- La tecnica di fact-finding più importante e più utilizzata
 - L'analista di sistemi raccoglie informazioni faccia a faccia
- Chi dovrebbe essere intervistato?
 - Dirigenti nelle prime fasi del progetto per ottenere un'ampia comprensione iniziale
 - Lo staff può fornire dettagli e aspetti specifici in seguito
 - Le questioni politiche possono essere importanti –può essere necessario intervistare persone influenti, anche se non sono troppo informate

Ruolo degli Use Case

- Gli Use Case esprimono e chiariscono i requisiti utente
- Scopo - definire l'interazione attesa tra utente e sistema
- Usare quella interazione per descrivere più compiutamente i requisiti funzionali
- Sono utilizzati estesamente nella fase di analisi, spesso una parte delle interviste agli utenti o delle sessioni JAD
- Gli Use case in forma testuale sono facili da comprendere per gli utenti
 - E gli use case UML ?
- Si inserisci facilmente nella creazione dei modelli dei processi e in quello dei dati

I casi d'uso

- Il caso d'uso rappresenta come un sistema interagisce con il suo ambiente
- Illustra le attività eseguite dagli utenti e le risposte del sistema
- Le attività producono alcuni risultati di output
- Ogni Use case descrive come un utente esterno attiva un evento a cui il sistema deve rispondere
- In una modellazione event-driven, tutto nel sistema può essere pensato come una risposta ad un certo evento di trigger

Elementi di uno Use Case

- Ogni Use Case ha un *nome* e *numero*, e una breve *descrizione*.
- Una *priorità* può essere assegnata per indicare l'importanza relativa
- L' *attore* si riferisce a una persona, un altro sistema, o un dispositivo hardware che interagisce con il sistema per raggiungere uno scopo utile
- Il *trigger* per lo use case – l'evento che causa l'inizio dello Use Case
- Gli eventi di trigger possono essere *esterni* o *temporali*

Use Case Basic Information

- Use Case base

Use Case Name: Request a chemical	ID: UC-2	Priority: High
Actor: Lawn Chemical Applicator (LCA)		
Description: The Lawn Chemical Applicator (LCA) specifies the lawn chemical needed for a job by entering its name or ID number. The system satisfies the request by reserving the quantity requested or the quantity available and notifying the Chemical Supply Warehouse of the pick-up.		
Trigger: A Lawn Chemical Applicator (LCA) needs a chemical for a job.		
Type: <input checked="" type="checkbox"/> External <input type="checkbox"/> Temporal		
Preconditions:		
1. The LCA identity is authenticated. 2. The LCA has necessary training and credentials on file. 3. The Chemical Supply datastore is up-to-date and on-line.		
Postconditions:		
1. The Lawn Chemical Request is stored in the Chemical Management System. 2. The Chemical Pick-up Authorization is produced for the LCA. 3. The Chemical Supply Warehouse is notified of the chemical pick-up. 4. Purchasing is notified of chemical outage.		

Use Case e i requisiti funzionali

- Gli Use case sono strumenti utili per chiarire i requisiti utente
- Gli Use case trasmettono solo il punto di vista dell'utente
- Trasformare la vista dell'utente nella vista dello sviluppatore attraverso i requisiti funzionali è uno dei contributi chiave del system analyst**
- I requisiti funzionali che vengono derivati, assieme alle specifiche che sono il risultato della progettazione, diranno allo sviluppatore di più su quello che il sistema deve fare

Requisiti funzionali dettagliati

- Il contenuto dell'Use case è utilizzato per creare requisiti funzionali più completi e descrittivi

- The system shall allow the LCA who is logged in to the Chemical Request system to request one or more chemicals.
- The system shall allow the LCA to specify a chemical by entering its ID number or name.
- The system shall notify the LCA if the chemical is no longer approved for use.
- The system will prompt the LCA for the quantity of the chemical needed.
- The system shall search the Chemical Supply datastore for the quantity available of the requested chemical and display the quantity available.
- The system shall prompt the user to confirm his request.
- When the request is confirmed, the system shall do the following as a single transaction:
 - Assign the next Chemical Request number to the Chemical Request, assign the current date and time to the Chemical Request, record the LCA's name and ID number on the request.
 - Update the amount available of the chemical by subtracting the quantity requested from the quantity available in the Chemical Supply datastore.
 - Print the Chemical Pick-up Authorization Notice for the LCA.
 - Send a message to the Chemical Supply Warehouse of the approved Chemical Pick-up.
 - Record the approved Chemical Request in the Chemical Request datastore, marked as "Pending Pick-up."
- The system shall prompt the LCA to exit the system or to make another chemical request.

Creare Use Case

- Identificare gli eventi a cui il sistema deve rispondere – sviluppare liste Event-Response
- Creare un modulo di Use case per gli eventi complessi
- Per ogni Use case:
 - Identificare i passi principali
 - Identificare gli elementi con ogni step principale (input e output)
 - Confermare lo Use case con gli utenti mediante il gioco di ruolo
- Revisionare i requisiti funzionali come necessario

Analisi dei processi business

Argomenti

- L'analisi dei processi business
- UML ed il suo ruolo
- **Modellazione dei BP con BPMN**
- Analisi del processo come successione di attività
- Analisi del processo come successione di casi d'uso
- Analisi delle entità che prendono parte ai processi
- Analisi delle interazioni tra gli elementi di un processo
- Un esempio completo di analisi
- Una visione d'insieme: il legame fra le viste e i pattern

Business Analysis (o analisi business)

Insieme di attività, conoscenze e tecniche necessarie per identificare le esigenze di business di un'organizzazione e determinare migliorie e soluzioni agli eventuali problemi esistenti.

Business Analysis (o analisi business) (2)

- Disciplina che consente di identificare le **esigenze di business** e determinare le **soluzioni ai problemi aziendali**.

Passaggio alla visione per processi

- **Business Process Management (BPM):** gestione ottimale di un processo, in modo da ridurre i suoi costi ed aumentare il valore da esso prodotto
- **Business Process Modeling:** analisi e modellazione del processo

Business (process) modeling

Attività di rappresentare i processi di una organizzazione per potere:

- analizzarli
- scoprirne eventuali difetti
- misurarne efficacia ed efficienza
- migliorarne il funzionamento

Business Architecture

“Una cianografia (figura disegnata nei contorni) dell’azienda (od organizzazione) che provvede una conoscenza comune della sua organizzazione e viene usata per allineare gli obiettivi strategici e le esigenze tattiche.”

“(The)Business Architecture defines the structure of the enterprise in terms of its governance structure, business processes, and business information. In defining the structure of the enterprise, business architecture considers customers, finances, and the ever-changing market to align strategic goals and objectives with decisions regarding products and services; partners and suppliers; organization; capabilities; and key initiatives.”

(OMG)

Business Architecture (2)

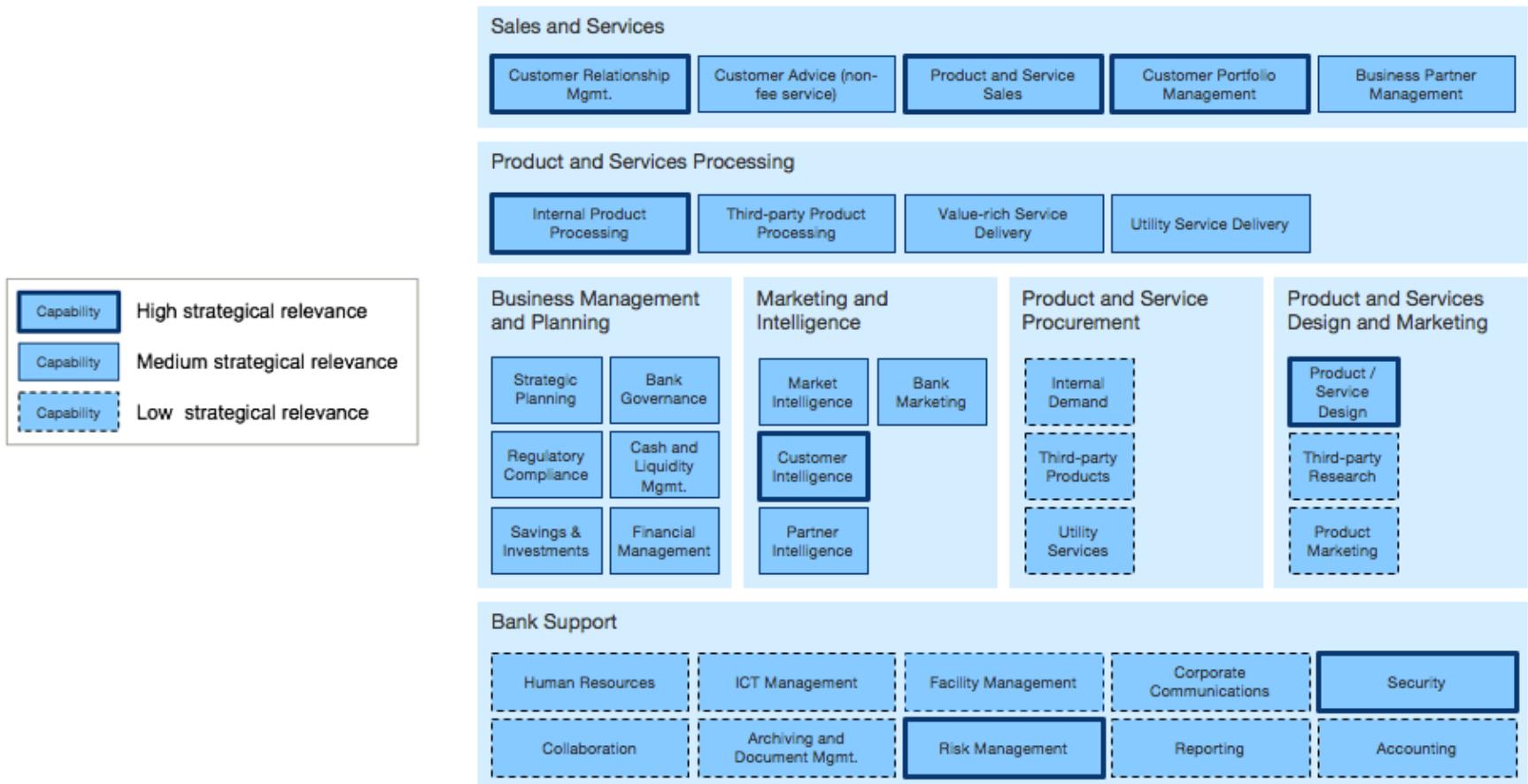
Definisce la struttura funzionale di una organizzazione in termini

- di **servizi business** e
- delle **informazioni business** per essi necessarie

Business Capability

Abilità che i servizi business hanno di provvedere le funzionalità business necessarie portando i risultati business attesi ed il valore ad essi associato

Esempio di business capabilities



Passaggio alla visione per processi

- **Business Process Improvement (BPI):** interventi di tipo incrementale e cioè volti al continuo e graduale miglioramento dei processi
- **Business Process Reengineering (BPR):** interventi di tipo radicale e cioè volti al completo ridisegno del processo qualora esso si manifesti del tutto inadeguato agli obiettivi da raggiungere

UML

- Unified Modeling Language
- Linguaggio unificato per la modellazione di concetti, entità, funzionalità, processi e relazioni che fra essi intercorrono
- Nasce nel 1997
- Unisce precedenti sintassi di modellazione (Booch, OMT, OOSE)
- Standard gestito dal consorzio OMG (<http://www.omg.org>)

UML come strumento di modellazione

Il linguaggio UML serve a descrivere, in modo grafico e “compatto”

- I requisiti utente (use case)
- Le componenti dei sistemi
- I dati in esse contenuti
- Le azioni da esse svolte
- Le relazioni che fra loro intercorrono (il processo in cui esse operano)

Evoluzione di UML

- Dopo il 1997 le estensioni standard di UML si susseguono
- Anche l'uso viene esteso alle varie fasi di realizzazione dei sistemi IT
- Attualmente i riferimenti sono lo standard 1.5 (per alcuni strumenti) e il 2.5 (per la maggioranza degli strumenti)

Usare UML

- Normalmente si usano programmi appositi, i Computer-Aided Software Engineering tool (CASE)
- Molti CASE sul mercato, sia open, sia commerciali
- Non sempre interscambio formati possibile
- Un formato standard: XMI (**XML Metadata Interchange**) con limiti

Usare UML: alcuni CASE

- IBM Rational Rose (commerciale)
- Gentleware Poseidon for UML (com.)
- Tigris ArgoUML (free)
- StarUML (free ma non più supportato)
- Visual Paradigm (free e com.)
- PlugIn di Eclipse: AmaterasUML (free)
- PlugIn di VS e di NetBeans
- ...

Obiettivo di UML

- Linguaggio di modellazione, semigrafico, utilizzabile da tecnici, da non tecnici e dalle macchine
- Un modello è una semplificazione della realtà che si ottiene
 - Riducendo le caratteristiche in esame
 - Considerando solo quelle utili al fine dell'analisi in corso
- Potenza espressiva nella documentazione

L'evoluzione logica di UML

- Nella catena di produzione dei sistemi IT, l'accento viene posto sempre più sulle fasi di analisi e progettazione
 - Una fase di analisi deve provvedere la conoscenza di dove si è (AS IS) e dove si vuole andare (TO BE)
- Uso di UML anche nel Business Modeling

Tipi di diagrammi UML: analisi

- dei casi d'uso - **Use Case Diagram**
- delle classi - **Class Diagram**
- degli oggetti – **Object Diagram**
- di sequenza - **Sequence Diagram**
- di collaborazione - **Collaboration** o **Communication Diagram**
- di transizione di stato - **Statechart Diagram**
- delle attività - **Activity Diagram**

Tipi di diagrammi UML: implementazione

- dei componenti - **Component Diagram**
- di distribuzione - **Deployment Diagram**

Diagrammi UML 2.x: strutturali

- **Package diagram**
- **Class o Structural diagram**
- **Object diagram**
- **Composite Structure diagram**
- **Component diagram**
- **Deployment diagram**
- **Profile diagram**

Diagrammi UML 2.x: comportamentali

- **Use Case diagram**
- **Activity diagram**
- **State Machine diagram**

Comportamentali di interazione

- **Communication diagram**
- **Sequence diagram**
- **Timing diagram**
- **Interaction Overview diagram**

Il concetto di Vista

Vista, traduzione dell’inglese view

- Un diagramma esprime solo alcune caratteristiche
 - definisce un modello che proietta e rende visibili solo alcune caratteristiche
 - Quelle che si vuole esaminare

L'uso dei diagrammi UML: analisi e design

- **Use Case Diagram:** per capire nei dettagli “cosa” il sistema deve fare
- **Class/Object Diagram:** per definire le entità fondamentali
- **Activity/Statechart Diagram:** per definire i processi fondamentali (fortemente “imparentati” con gli Use Case Diagram)
- **Collaboration Diagram:** per definire le interazioni fra le entità fondamentali
- **Sequence Diagram:** per definire la sequenza delle interazioni fra entità

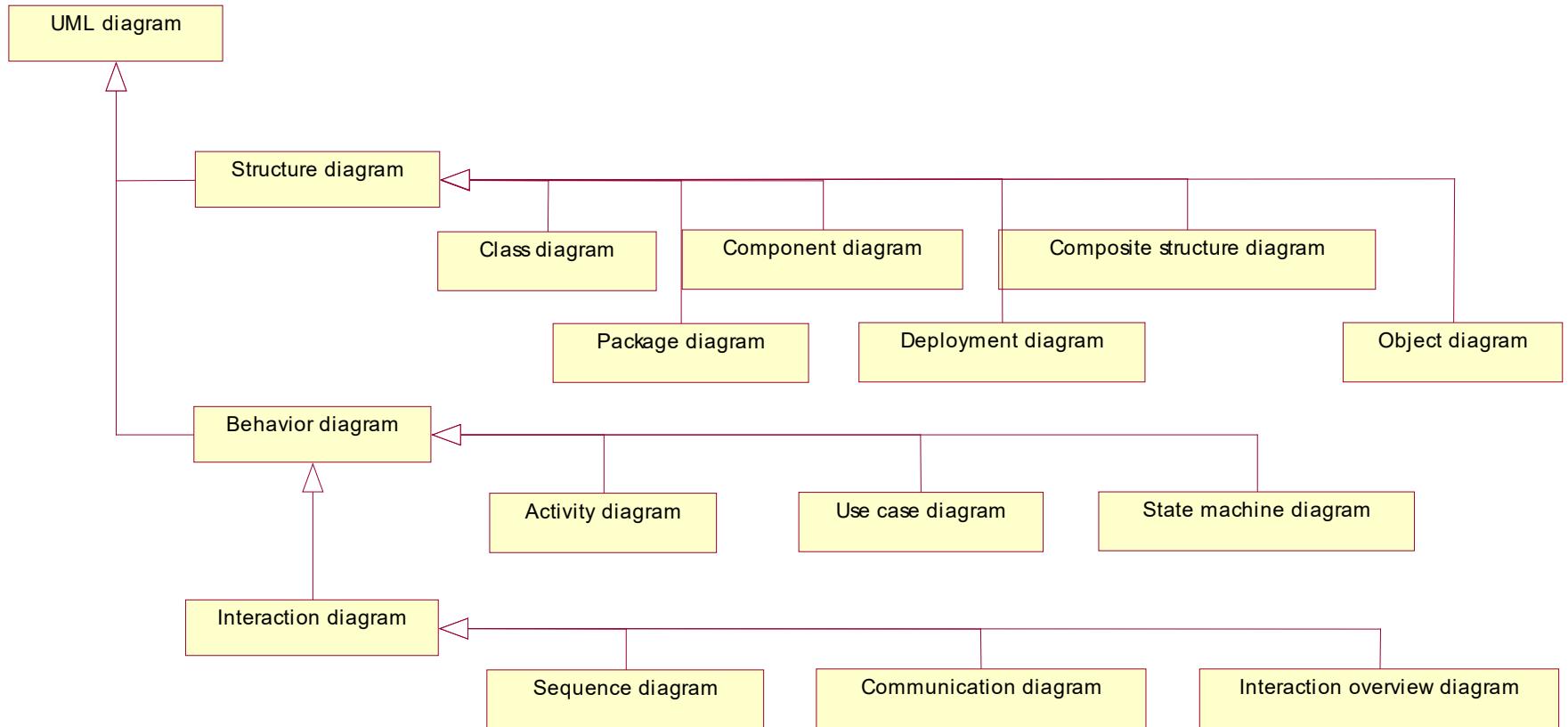
L'uso dei diagrammi UML: implementazione/installazione

- **Component Diagram:** per definire nei dettagli quali parti compongono il sistema/prodotto software finito e le loro relazioni
- **Deployment Diagram:** per definire dove le varie parti di un programma devono essere poste in un'architettura distribuita (es. client-server)

Riassumendo

- Modelli Funzionali/User's view
 - Use case diagram
- Modelli di strutture (statici)/Structural view
 - Class diagram
- Modelli dinamici o di comportamento/Behavioral View
 - Statechart diagram
 - Sequence diagram
 - Activity diagram

Riassumendo



Diagrammi UML 2.x: meno usati

- **Package diagram**
- **Composite Structure diagram**
- **Profile diagram**
- **Timing diagram**
- **Interaction Overview diagram**

Uso di UML: IT e Business Modeling

- Il sistema informativo è l'insieme delle risorse umane, materiali, organizzative (regole) che gestisce e distribuisce le informazioni entro l'azienda
- Il sistema informatico è l'insieme degli strumenti IT di un sistema informativo
- Serve un linguaggio chiaro comune a tutti i componenti del sistema informativo

Uso di UML: Business Modeling

- Usare le stesse notazioni di UML per
 - Esprimere i concetti del business
 - Per uso diretto (per uso di analisi business)
 - Per uso indiretto (in vista di una successiva implementazione di sistemi IT di ausilio al business)

Uso di UML: Business Modeling

- Usare UML in tutto il sistema informativo, come linguaggio comune
- Il processo di sviluppo software usa lo stesso linguaggio sin dalle prime fasi dell'analisi
- Comunicazione con fornitori IT con linguaggio comune

Uso di UML: Business Modeling

- Metodologie unificate
 - RUP (Rational Unified Process) di Rational (<http://www.rational.com>)
 - Scuola scandinava (Eriksson e Penker, *Business Modeling with UML – Business Patterns at Work*, 2000)
- Altri approcci
 - EbXML (Electronic business eXtensible markup language) ISO 15000

Vantaggi di UML nel Business Modeling

L'uso di UML consente di focalizzare l'attenzione

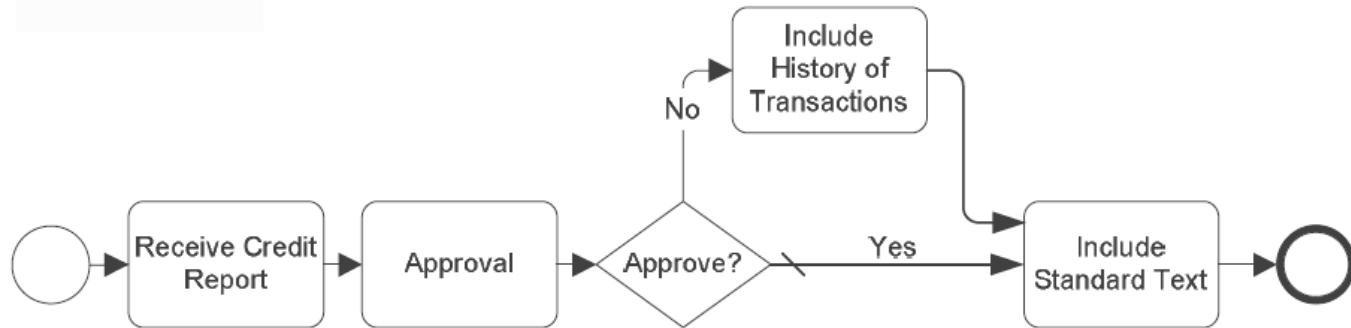
- sui vari elementi costitutivi un processo
- sui vari punti di vista che si possono seguire

BPMN: Business Process Model and Notation

- Evoluzione di UML for Business
- Inserisce tutti gli elementi necessari per l'analisi dettagliata dei processi
- Completando alcune carenze di UML
- Basato sul punto di vista del Business Analyst e dei suoi clienti
- Standard 2.0 del 2011 di OMG
- Pubblicato anche come ISO/IEC 19510:2013

Cos'è BPMN

- BPMN è una notazione basata su diagrammi di flusso per la definizione di processi business

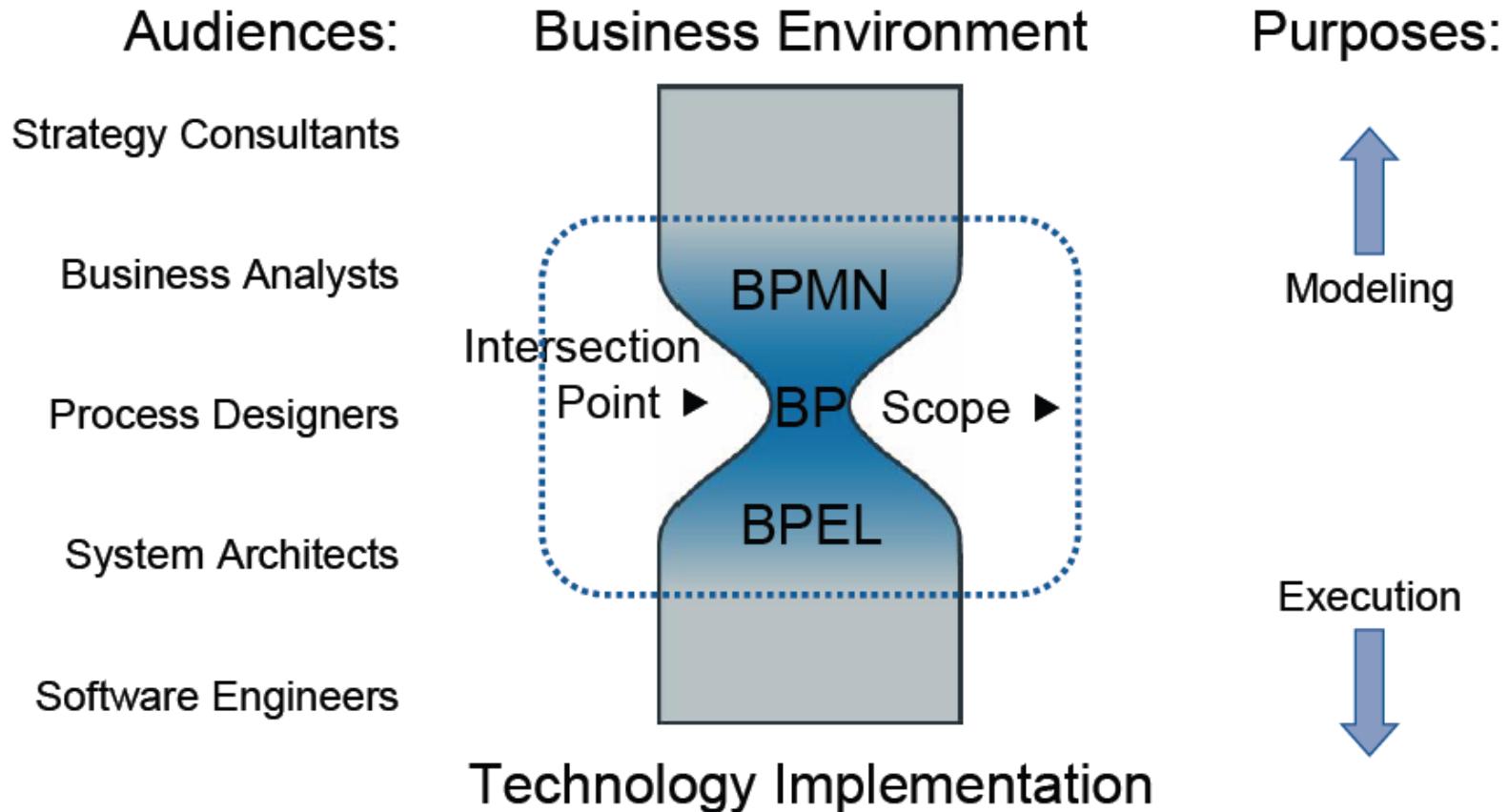


- BPMN è un insieme di oggetti grafici e regole che definiscono le possibili connessioni tra gli oggetti
- BPMN deriva da un accordo tra diversi produttori di tool di modellazione, ognuno con una propria notazione, per utilizzarne una condivisa a beneficio della comprensione e dell'apprendimento degli utenti finali

BPMN e BPEL

- BPMN fornisce un meccanismo per la generazione di un processo business **eseguibile**, a partire dalla notazione a livello business
- Un processo Business sviluppato da un analista business può essere direttamente applicato a un engine BPEL invece di ricorrere a interpretazioni umane e a traduzioni in altri linguaggi
- Nella realtà le differenze di livello e semantiche possono porre difficoltà e richiedere lavoro aggiuntivo

BPMN e BPEL



Esempio di BPEL

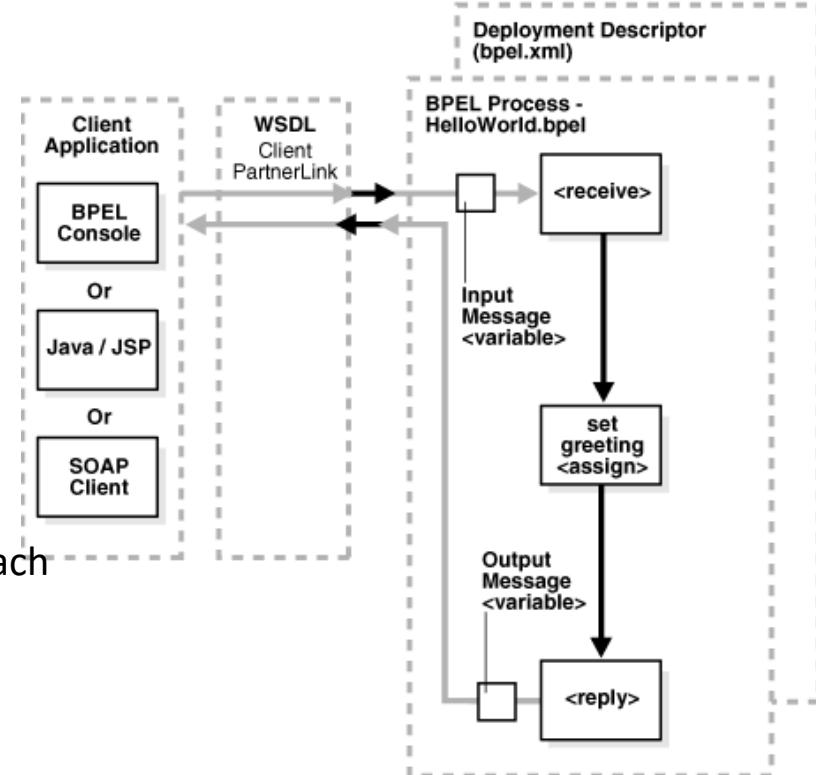
The Hello World application is an example of a BPEL process that accepts input, manipulates the input, returns an output through a synchronous reply.

The BPEL process has the following components:

- a client application: the input for the BPEL process comes from a client application that can be realized in multiple technologies
 - BPEL Console
 - Or Java / JSP
 - Or SOAP Client
- a partner link: the partner link connects the client application to the BPEL process.

It defines the role of each party, the types of data each accepts and returns, and so on.

The WSDL file contains the definition of the partner links



https://docs.oracle.com/cd/B14099_19/integrate.1012/b14448/hello.htm

Esempio di BPEL

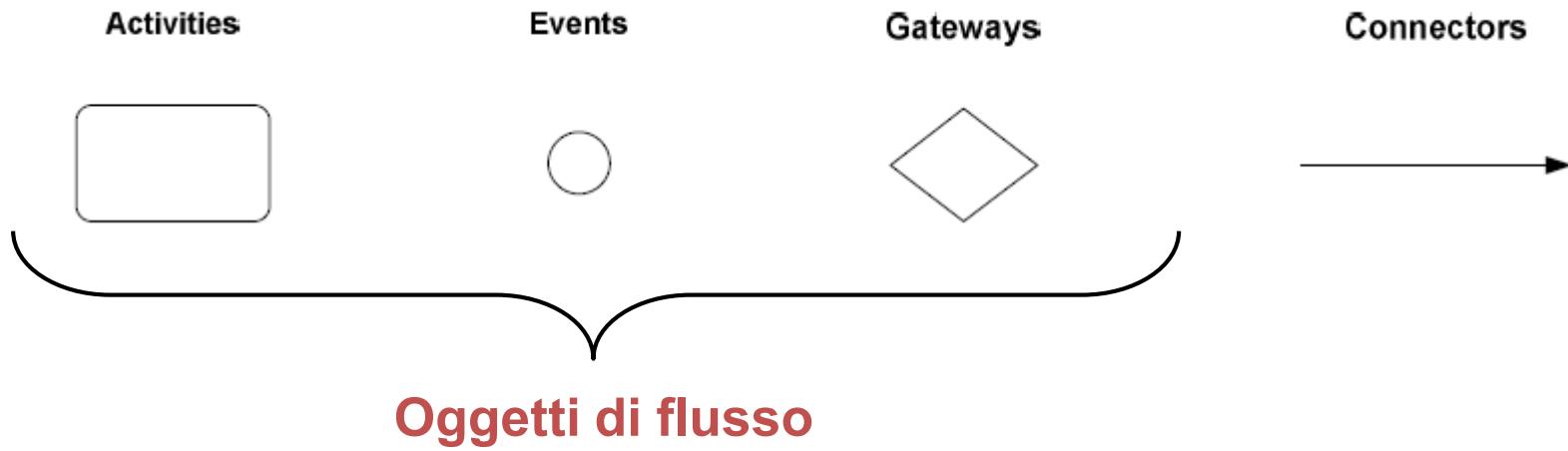
```
<!-- HelloWorld BPEL Process -->
<process name="HelloWorld"
targetNamespace="http://samples.otn.com/helloworld"
suppressJoinFailure="yes"
xmlns:tns="http://samples.otn.com/helloworld"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
>
<!-- List of services participating in this BPEL process -->
<partnerLinks>
<!--
The 'client' role represents the requester of this service. It is used for
callback.

The location and correlation information associated with the
client role are automatically set using WS-Addressing.

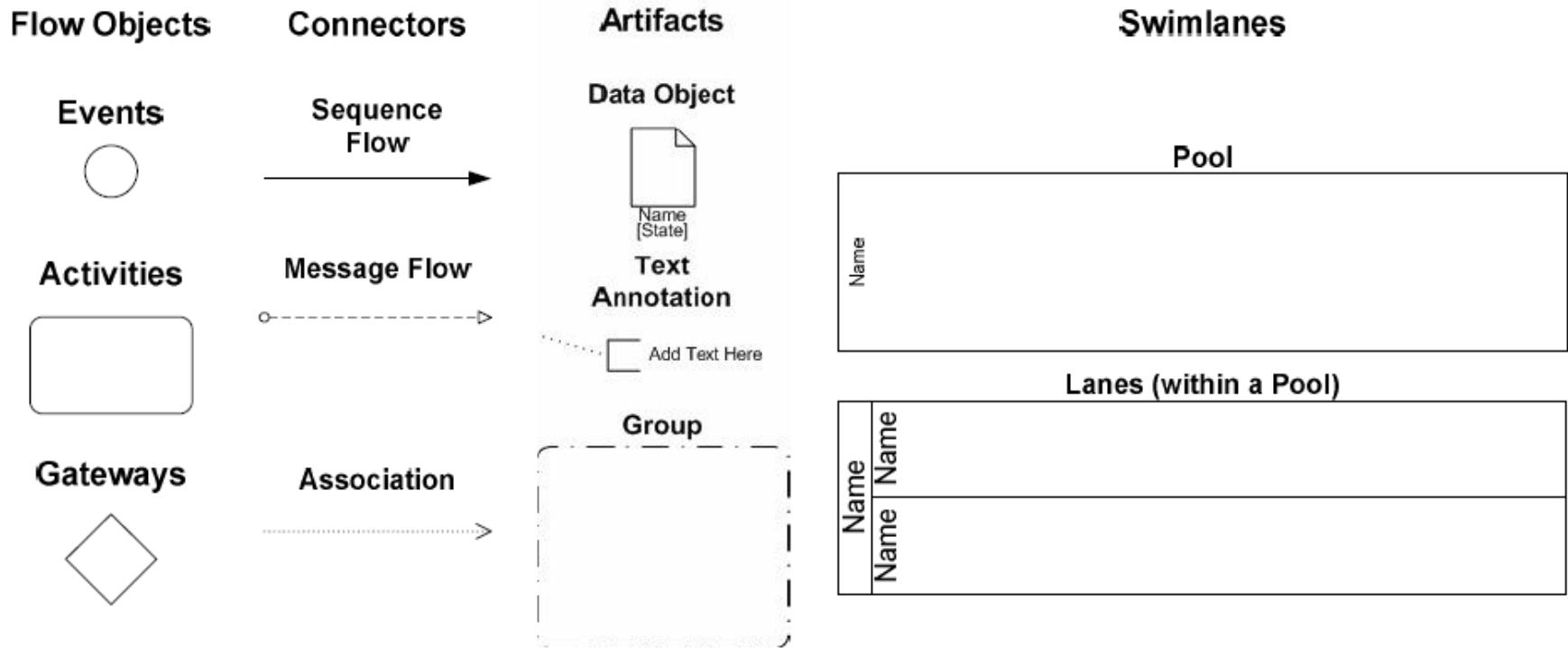
-->
<partnerLink name="client"
partnerLinkType="tns:HelloWorld"
myRole="HelloWorldProvider"
partnerRole="HelloWorldRequester"
/>
</partnerLinks>

<!-- List of messages and XML documents used as part of this BPEL process -->
<variables>
<!-- Reference to the message passed as input during initiation -->
<variable name="input"
messageType="tns:HelloWorldRequestMessage"/>
<!-- Reference to the message that will be sent back to the requestor during callback -->
<variable name="output"
messageType="tns:HelloWorldResultMessage"/>
</variables>
<!-- Orchestration Logic -->
<sequence>
<!-- Receive input from requestor.
Note: This maps to operation defined in HelloWorld.wsdl
-->
<receive name="receiveInput" partnerLink="client"
portType="tns:HelloWorld"
operation="initiate" variable="input"
createInstance="yes"/>
<!-- Generate content of output message based on the content of the input message. -->
<assign>
<copy>
<from expression="concat('Hello ',bpws:getVariableData('input','payload','/tns:name'))"/>
<to variable="output" part="payload" query="/tns:result"/>
</copy>
</assign>
<!-- Asynchronous callback to the requester.
Note: the callback location and correlation id is transparently handled using WS-addressing. -->
<invoke name="replyOutput"
partnerLink="client"
portType="tns:HelloWorldCallback"
operation="onResult"
inputVariable="output"
/>
</sequence>
</process>
```

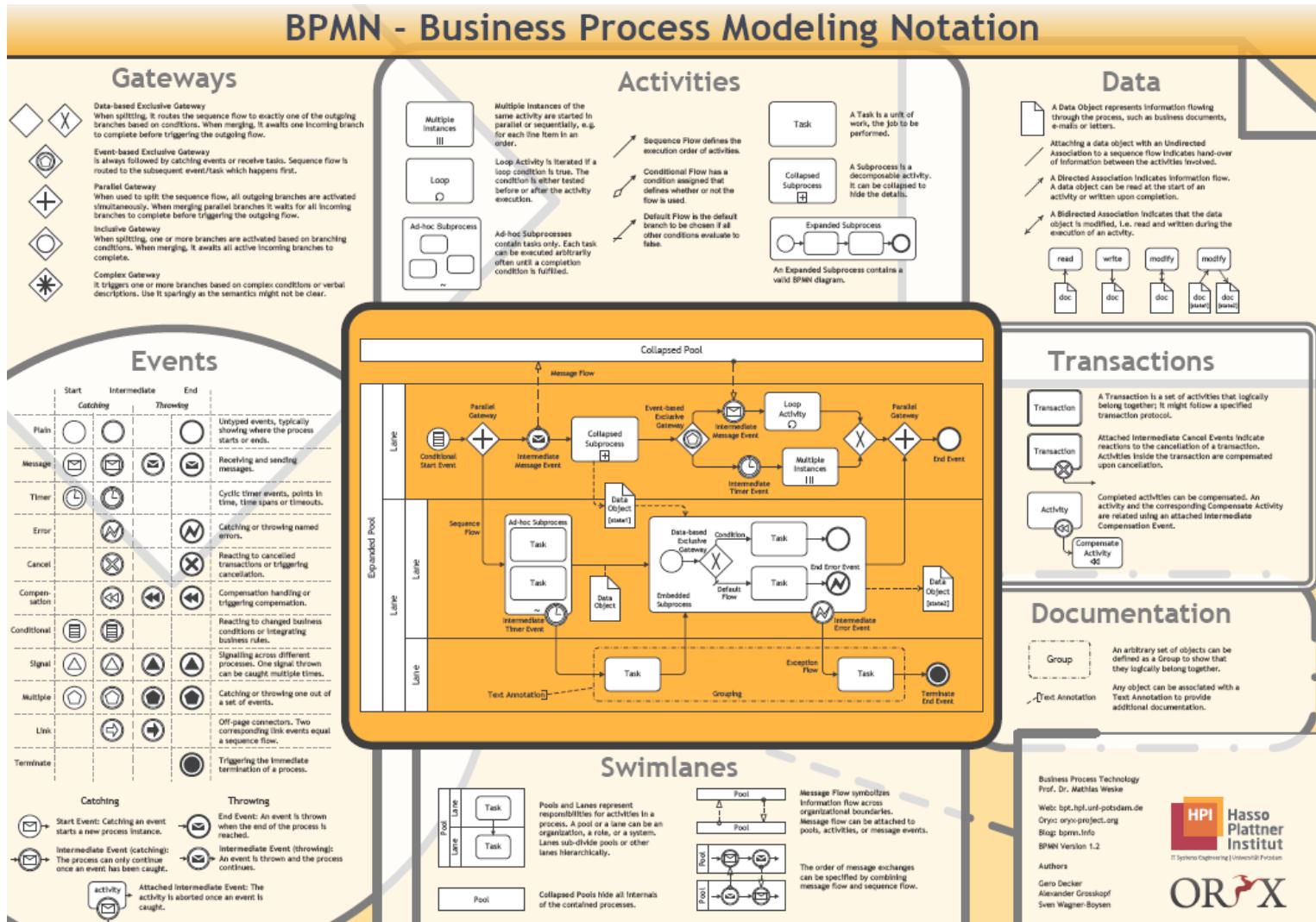
Elementi di base di BPMN



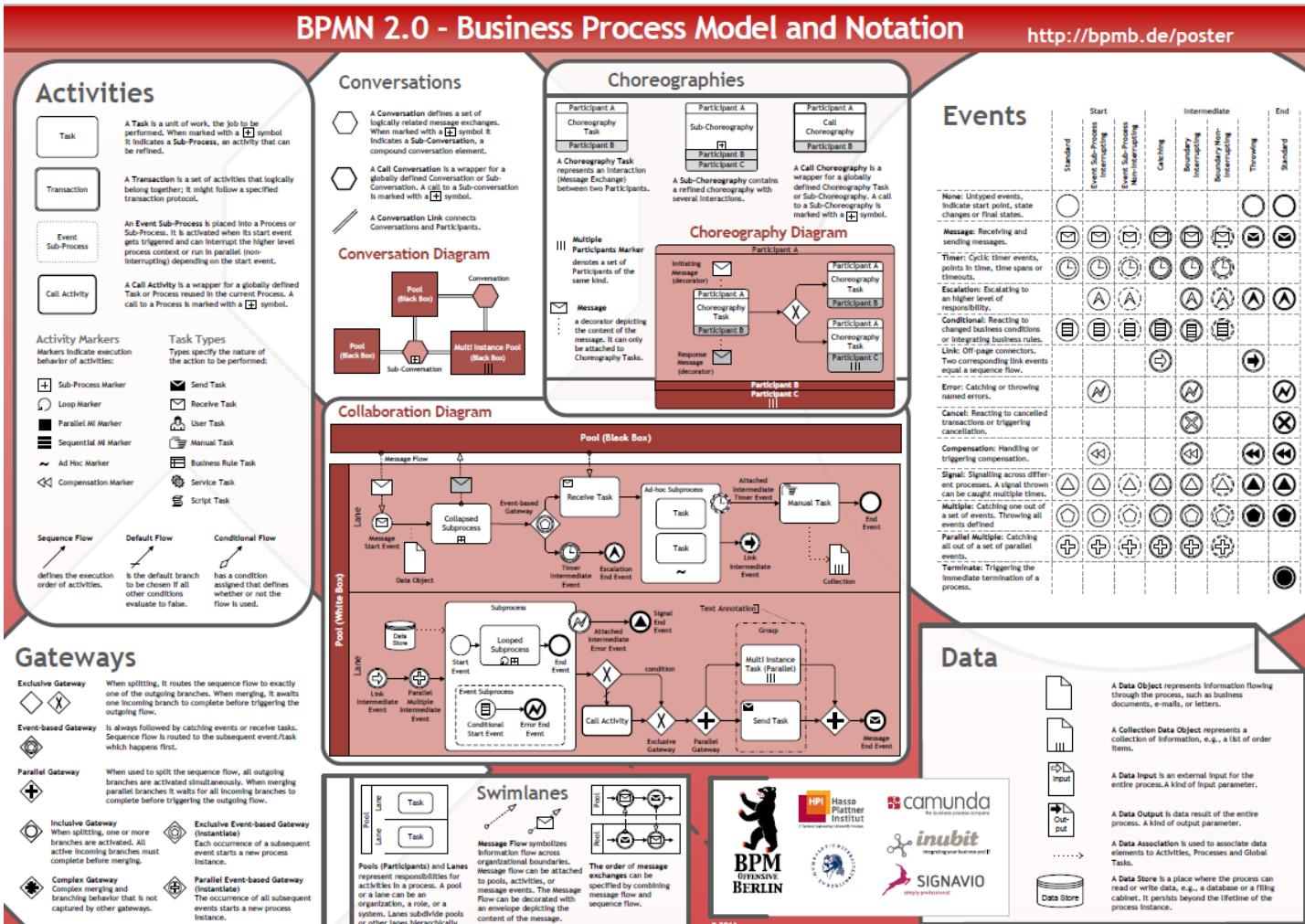
Elementi di base e aggiuntivi



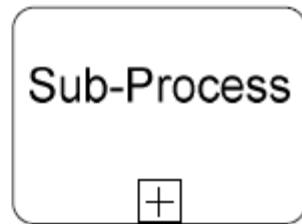
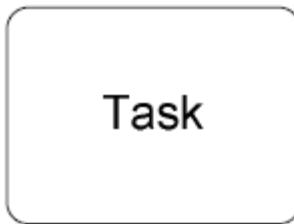
Il poster BPMN 1.0



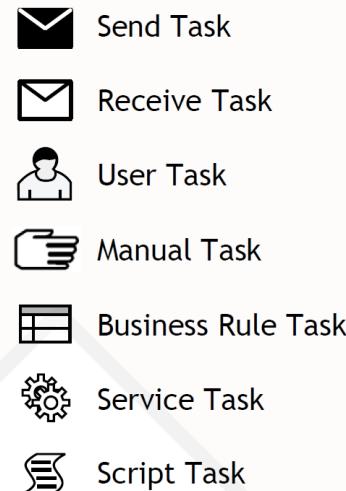
Il poster BPMN 2.0



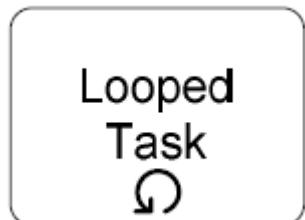
Attività



- Un'attività è un'unità di lavoro eseguito entro un processo business
- Un'attività può essere atomica (*Task*) o composta (*Sub-Process*)



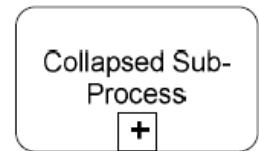
Tipi di Task



- Può essere eseguita una sola volta o avere cicli interni

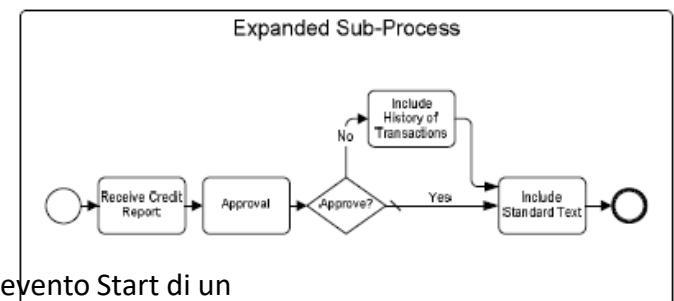
Sottoprocessi

- I sottoprocessi abilitano lo sviluppo gerarchico del Processo
- Un Sub-process è un'attività composta che è inclusa entro un processo
 - nella versione collassata del sottoprocesso (con il simbolo +) i dettagli non sono visibili nel diagramma
 - nella versione espansa i dettagli del sottoprocesso sono visibili



Ci sono due tipi di sottoprocessi:

- embedded (entro un processo)
- indipendente (globale e riusabile)



L'evento Start di un sottoprocesso deve essere di tipo **None**

Attività

Transaction

Event
Sub-Process

Call Activity

A **Transaction** is a set of activities that logically belong together; it might follow a specified transaction protocol.

An **Event Sub-Process** is placed into a Process or Sub-Process. It is activated when its start event gets triggered and can interrupt the higher level process context or run in parallel (non-interrupting) depending on the start event.

A **Call Activity** is a wrapper for a globally defined Task or Process reused in the current Process. A call to a Process is marked with a  symbol.

Eventi

Start



Intermediate



End



- Un evento è qualcosa che «accade» durante il corso di un processo business
- Gli eventi influenzano il flusso del processo e normalmente hanno un trigger o un risultato.
Possono iniziare, interrompere o arrestare il flusso
- Il tipo di bordo determina il tipo di evento

Eventi in BPMN 1.0

Gli eventi Start indicano dove/quando il processo inizierà

None



Message



Timer



Rule



Multiple



Gli eventi Intermedi avvengono durante il processo (possono anche essere attaccati ad un'attività - interrupt)

None



Message



Timer



Error



Compensation



Rule



Link



Multiple



Gli eventi End indicano dove il processo terminerà

None



Message



Error



Compensation



Signal



Terminate



Multiple



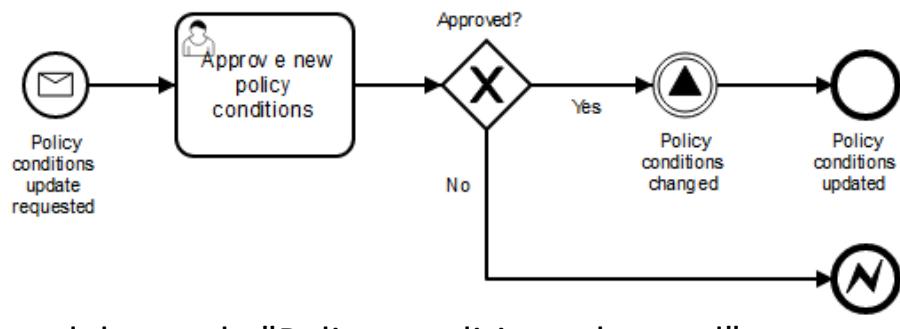
Eventi in BPMN 2.0

Events

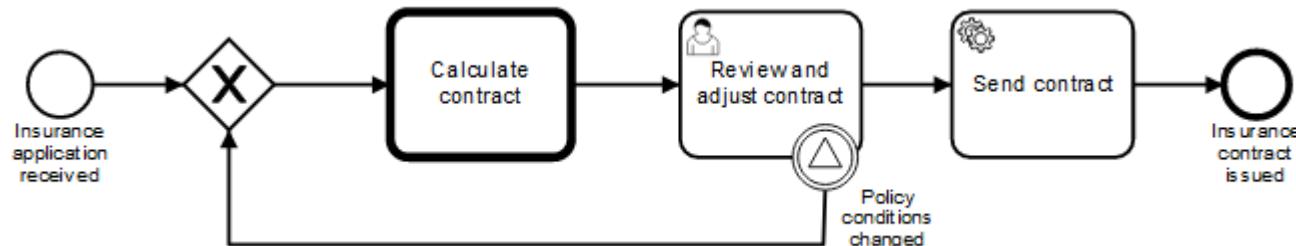
	Standard	Start	Event Sub-Process Interrupting	Event Sub-Process Non-Interrupting	Catching	Boundary Interrupting	Intermediate Non-Interrupting	End
Standard								
None: Untyped events, indicate start point, state changes or final states.								
Message: Receiving and sending messages.								
Timer: Cyclic timer events, points in time, time spans or timeouts.								
Escalation: Escalating to an higher level of responsibility.								
Conditional: Reacting to changed business conditions or integrating business rules.								
Link: Off-page connectors. Two corresponding link events equal a sequence flow.								
Error: Catching or throwing named errors.								
Cancel: Reacting to cancelled transactions or triggering cancellation.								
Compensation: Handling or triggering compensation.								
Signal: Signalling across different processes. A signal thrown can be caught multiple times.								
Multiple: Catching one out of a set of events. Throwing all events defined.								
Parallel Multiple: Catching all out of a set of parallel events.								
Terminate: Triggering the immediate termination of a process.								

Evento Signal

- Gli eventi Signal sono eventi che referenziano un segnale con nome. Un segnale è un evento di scope globale (semantica broadcast) ed è consegnato a tutti gli handler attivi



Un processo ricevitore del segnale "Policy conditions changed":



Gateway

I gateway sono usati per controllare come i flussi di sequenza interagiscono, convergendo o divergendo entro un processo

Importante:

i gateway non sono decisioni,
dirigono soltanto il flusso.

Il risultato di una decisione deve essere determinato da un'attività che precede i gateway

Exclusive
Data-Based

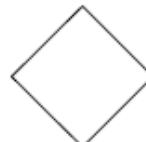
Data-Based

Event-Based

Inclusive

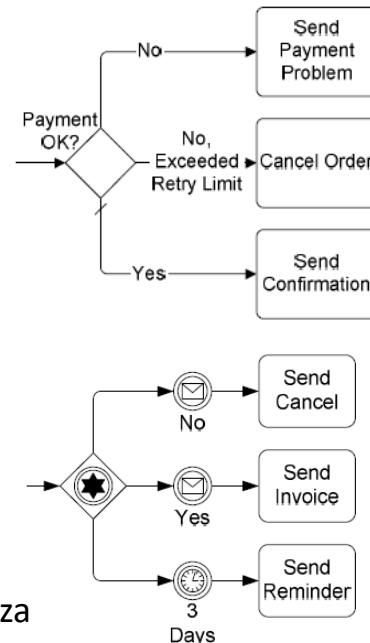
Complex

Parallel



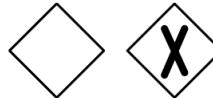
Non obbligatorio per suddividere il flusso ma consigliato per chiarezza

Flussi alternativi basati ad es.
su una condizione o sul
messaggio ricevuto



Gateway

Exclusive Gateway



When splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it awaits one incoming branch to complete before triggering the outgoing flow.

Event-based Gateway



Is always followed by catching events or receive tasks. Sequence flow is routed to the subsequent event/task which happens first.

Parallel Gateway



When used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow.



Inclusive Gateway

When splitting, one or more branches are activated. All active incoming branches must complete before merging.



Exclusive Event-based Gateway (instantiate)

Each occurrence of a subsequent event starts a new process instance.



Complex Gateway

Complex merging and branching behavior that is not captured by other gateways.



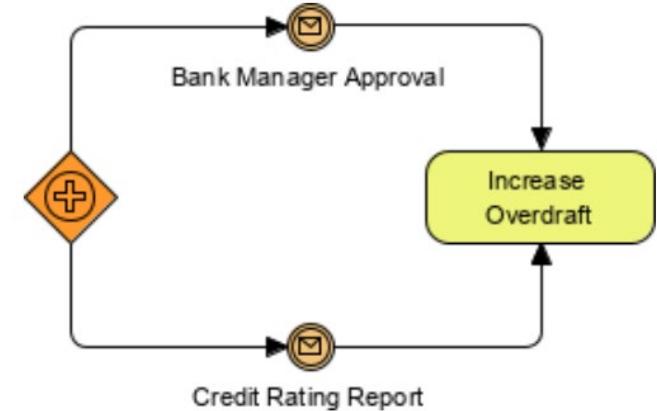
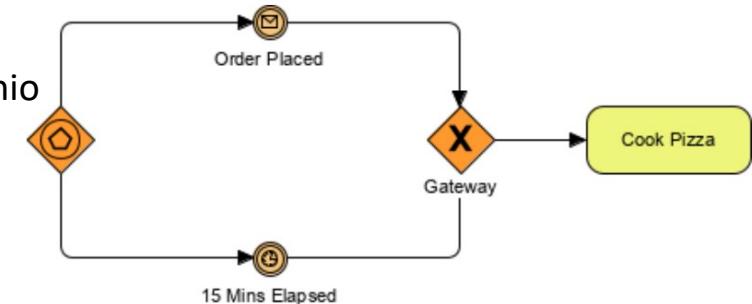
Parallel Event-based Gateway (instantiate)

The occurrence of all subsequent events starts a new process instance.

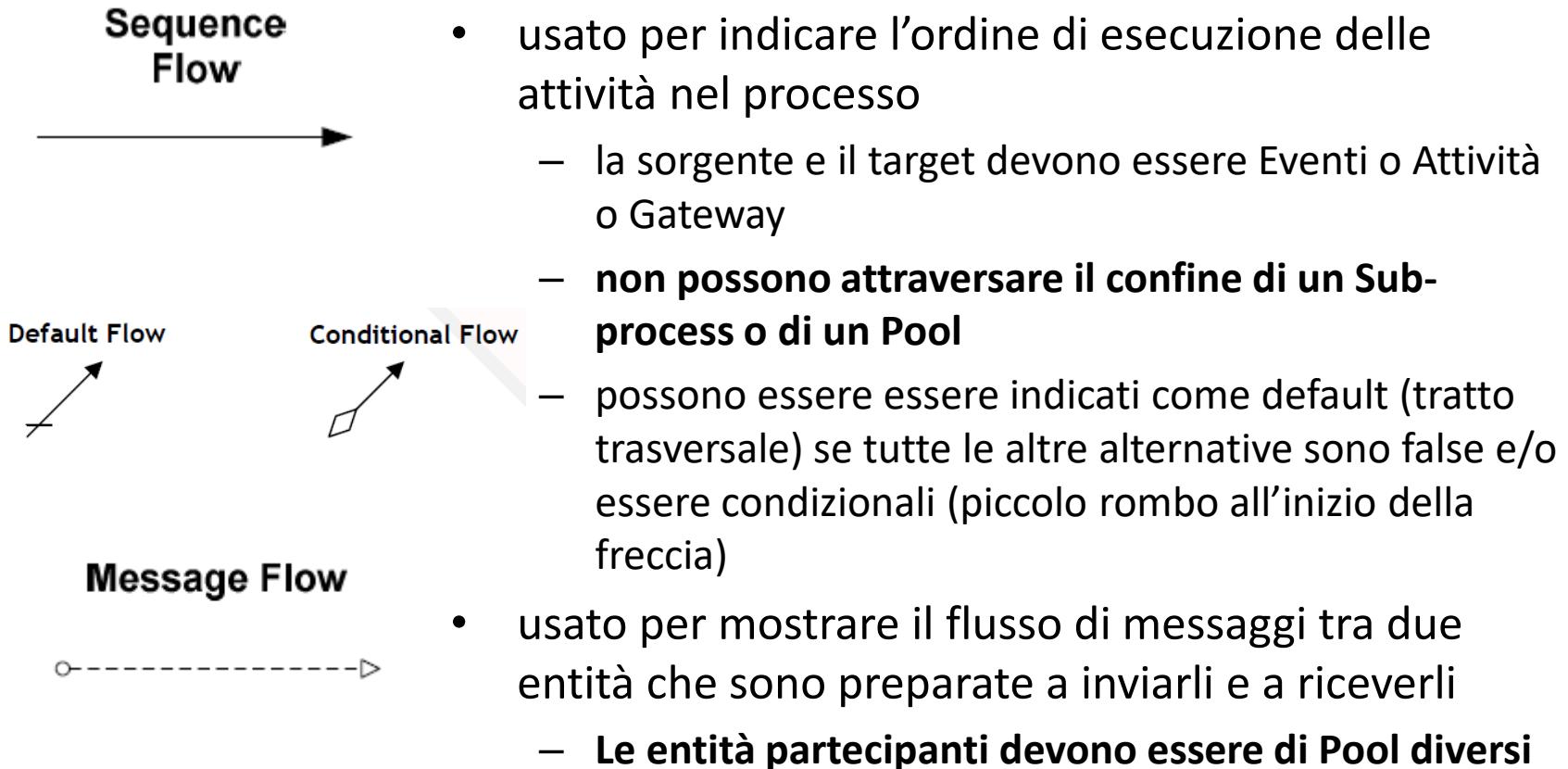


Exclusive/Parallel Event-Based Gateway

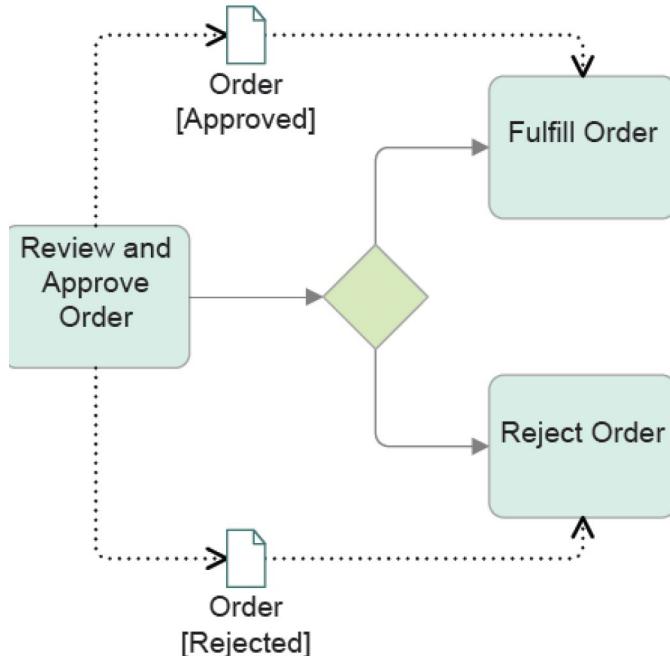
- Il gateway basato sugli eventi può essere utilizzato anche per istanziare un processo. In questo caso, l'icona del Gateway esclusivo basato su eventi presenta un solo cerchio all'interno del diamante. Quando viene utilizzato per avviare un processo, **l'Exclusive Event-Based Gateway** consente di avviare il processo in diversi modi, in base all'evento che lo attiva.
- Un **Parallel Event-Based Gateway** è simile a un gateway parallelo, dato che consente di eseguire più processi contemporaneamente. È importante notare che, sebbene il **Parallel Event-Based Gateway** consenta il passaggio di più eventi e l'avvio della parte corrispondente del processo, esso non attende l'arrivo di tutti gli eventi. Ciò significa che non attende e sincronizza gli eventi prima che sia consentito l'avvio di ciascun percorso di elaborazione. I gateway di eventi possono essere utilizzati per istanziare un processo. Per impostazione predefinita, l'attributo *Instantiate* del gateway è *False*, ma se viene impostato su *True*, il processo viene istanziato quando viene attivato il primo evento della configurazione del gateway. In questo esempio, se viene attivato l'evento Approvazione del gestore della banca, verrà eseguito il processo Aumento dello scoperto (*Increase Overdraft*).



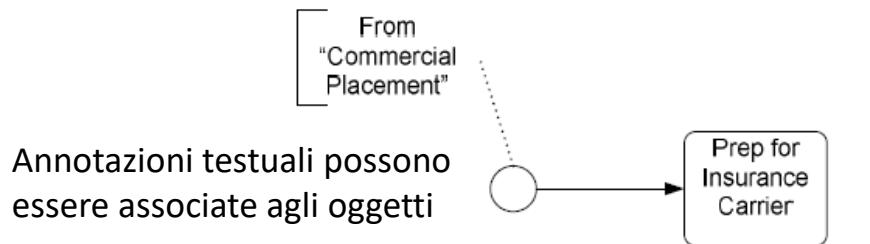
Connettori di flusso



Artefatti



In BPMN il data flow è **disaccoppiato** dal Sequence Flow



Annotazioni testuali possono essere associate agli oggetti

- Gli Artefatti (Data, Group e Annotation) possono rendere più chiari i diagrammi
- I Data Object sono utilizzati per mostrare come i dati entrano ed escono dalle Attività



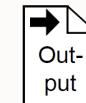
A **Data Object** represents information flowing through the process, such as business documents, e-mails, or letters.



A **Collection Data Object** represents a collection of information, e.g., a list of order items.



A **Data Input** is an external input for the entire process. A kind of input parameter.



A **Data Output** is data result of the entire process. A kind of output parameter.



A **Data Association** is used to associate data elements to Activities, Processes and Global Tasks.

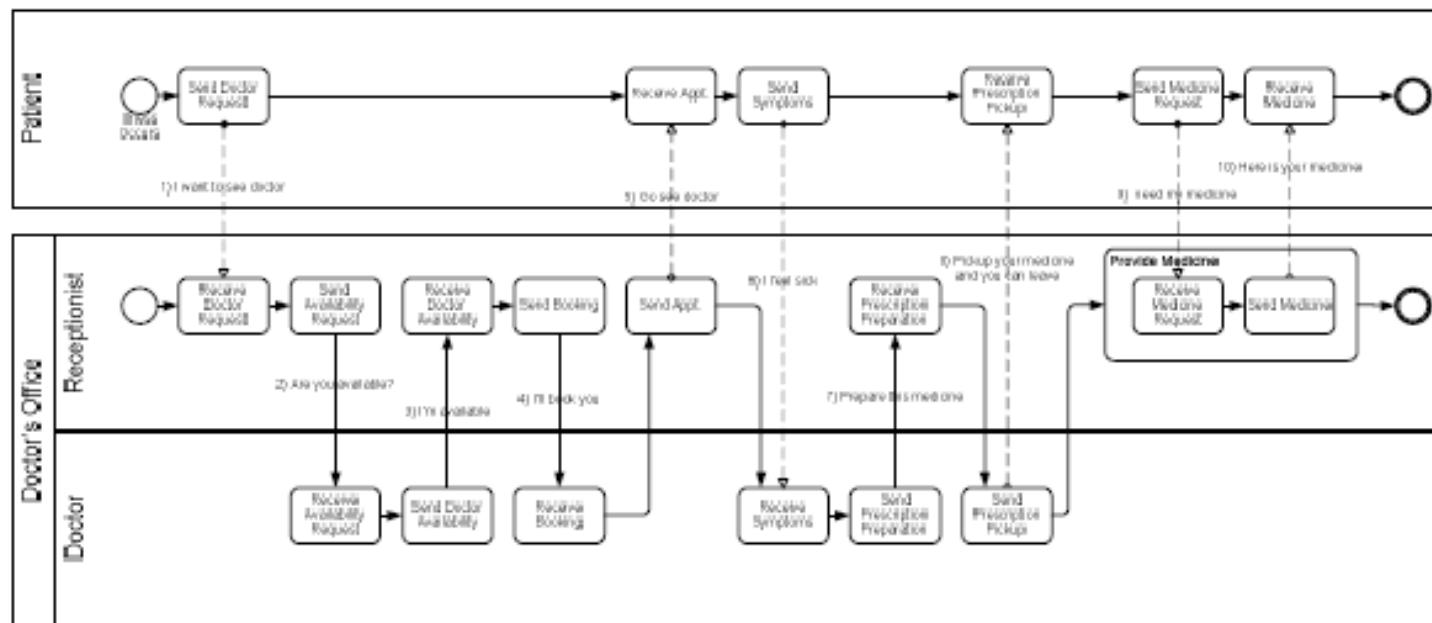
A **Data Store** is a place where the process can read or write data, e.g., a database or a filing cabinet. It persists beyond the lifetime of the process instance.

Swimlane

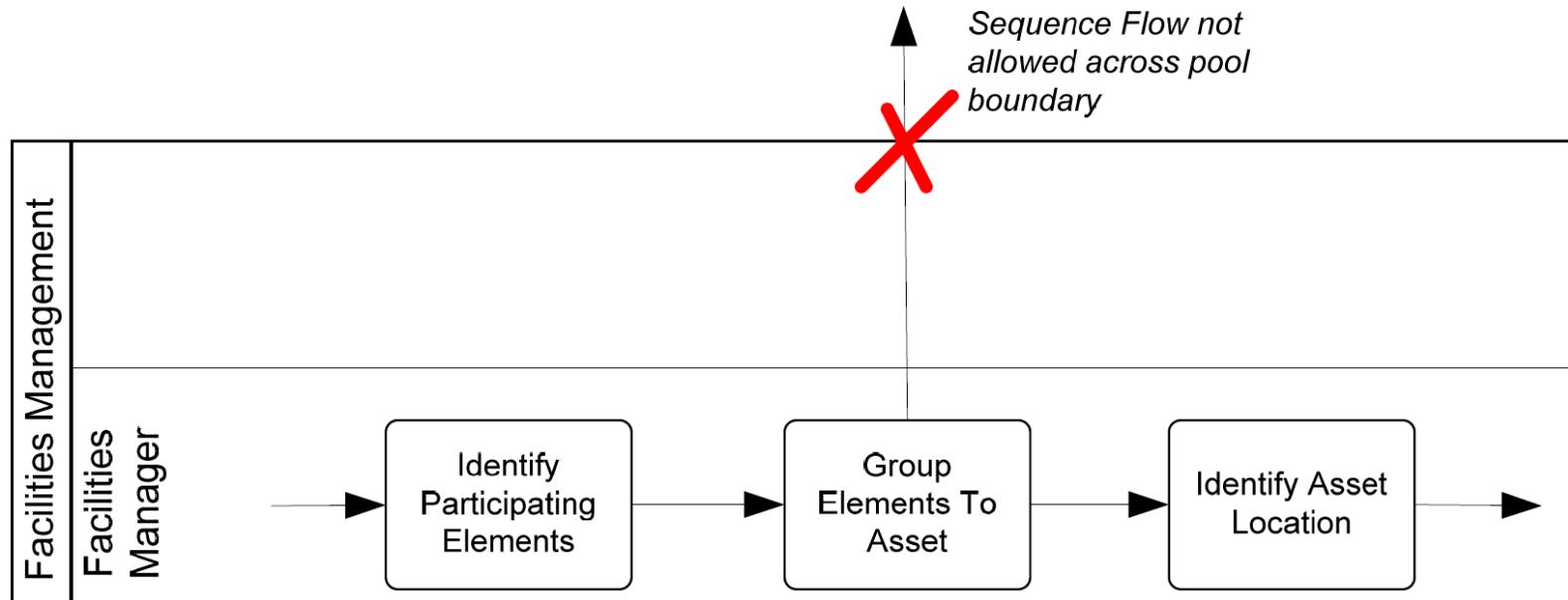
BPMN utilizza le swimlane per partizionare e/o organizzare le Attività:

Pool : rappresenta i partecipanti (ruoli o entità) in un Business Diagram interattivo (B2B)

Lane : rappresenta sotto-partizioni (spesso ruoli organizzativi) per gli oggetti all'interno di un Pool



Flusso di sequenza e pool



Un flusso di sequenza non può passare da un pool ad un altro ma può passare da un lane ad un altro dello stesso pool

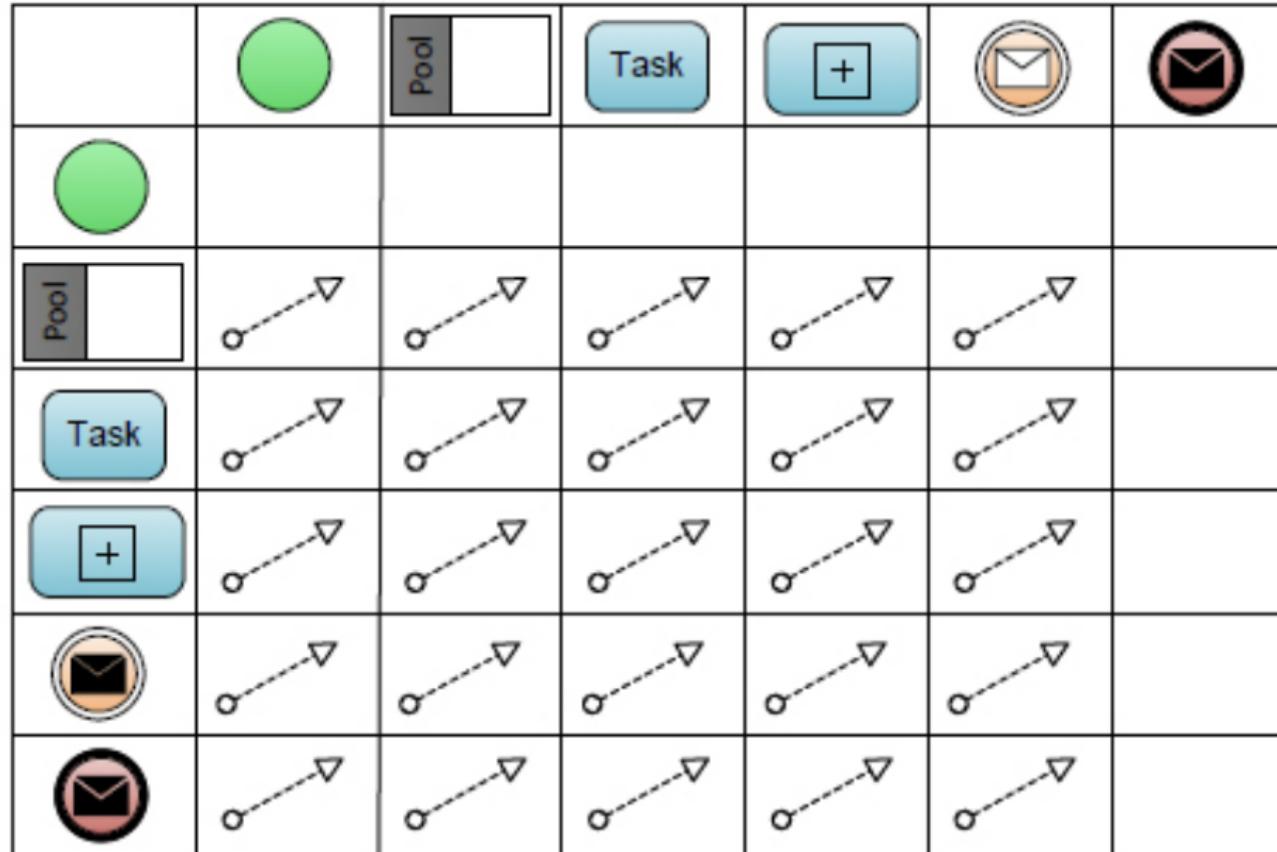
Regole di connessione per Message Flow

From\To	✉	Name Pool			✉	✉
✉						
Name Pool		↗	↗	↗	↗	
		↗	↗	↗	↗	
+		↗	↗	↗	↗	
✉		↗	↗	↗	↗	
✉		↗	↗	↗	↗	

A red dashed line with a question mark is drawn vertically through the second column of the table, indicating a specific connection rule between the 'From' and 'To' columns.

Tab. 7.4 del documento di specifica formale di BPMN 2.0

Regole di connessione per Message Flow



Categorie di processi

BPMN cerca di supportare tre categorie di processi:

- **Orchestration** : definisce una prospettiva di un singolo coordinamento, tipicamente un business specifico o la vista di un processo di un'organizzazione (quindi entro un singolo pool)
- **Choreography** : definisce il comportamento atteso (contratto o protocollo) della sequenza di interazioni tra partecipanti che possono essere ruoli di business (ad es. spedizioniere) oppure una specifica entità di business (ad es. DHL o BRT).
- **Collaboration** : mostra partecipanti e le loro interazioni (può includere una coreografia e una o più orchestrazioni)

Process Modeling Conformance

Per essere conforme un'implementazione deve supportare:

- The **BPMN** core elements, which include those defined in the *Infrastructure*, *Foundation*, *Common*, and *Service* packages (see Clause 8).
- **Process** diagrams, which include the elements defined in the **Process**, **Activities**, *Data*, and *Human Interaction* packages (see Clause 10)
- **Collaboration** diagrams, which include **Pools** and **Message Flow** (see Clause 9).
- **Conversation** diagrams, which include **Pools**, **Conversations**, and **Conversation Links** (see Clause 9).

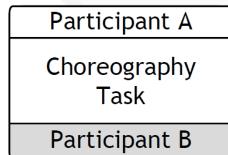
Dal documento di specifica formale di BPMN 2.0.2

Conformance sub-classes

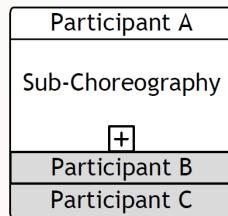
- **Descriptive** is concerned with visible elements and attributes used in high-level modeling. It should be comfortable for analysts who have used BPA flowcharting tools.
- **Analytic** contains all of **Descriptive** and in total about half of the constructs in the full **Process Modeling Conformance** Class. It is based on experience gathered in BPMN training and an analysis of user-patterns in the Department of Defense Architecture Framework and planned standardization for that framework. Both **Descriptive** and **Analytic** focus on visible elements and a minimal subset of supporting attributes/elements.
- **Common Executable** focuses on what is needed for executable process models.

Dal documento di specifica formale di BPMN 2.0.2

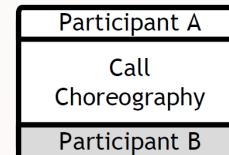
Coreografia in BPMN 2.0



A **Choreography Task** represents an Interaction (Message Exchange) between two Participants.



A **Sub-Choreography** contains a refined choreography with several Interactions.



A **Call Choreography** is a wrapper for a globally defined Choreography Task or Sub-Choreography. A call to a Sub-Choreography is marked with a **[+]** symbol.

III Multiple Participants Marker

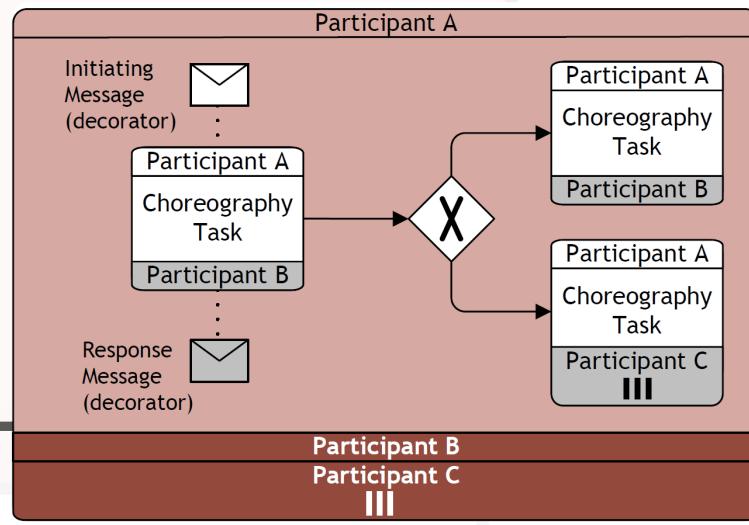
denotes a set of Participants of the same kind.



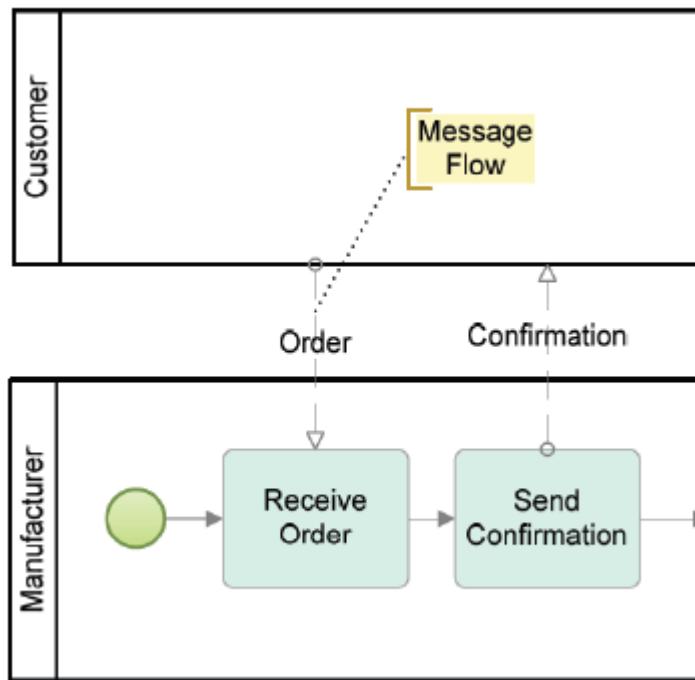
Message

- : a decorator depicting the content of the message. It can only be attached to Choreography Tasks.

Choreography Diagram



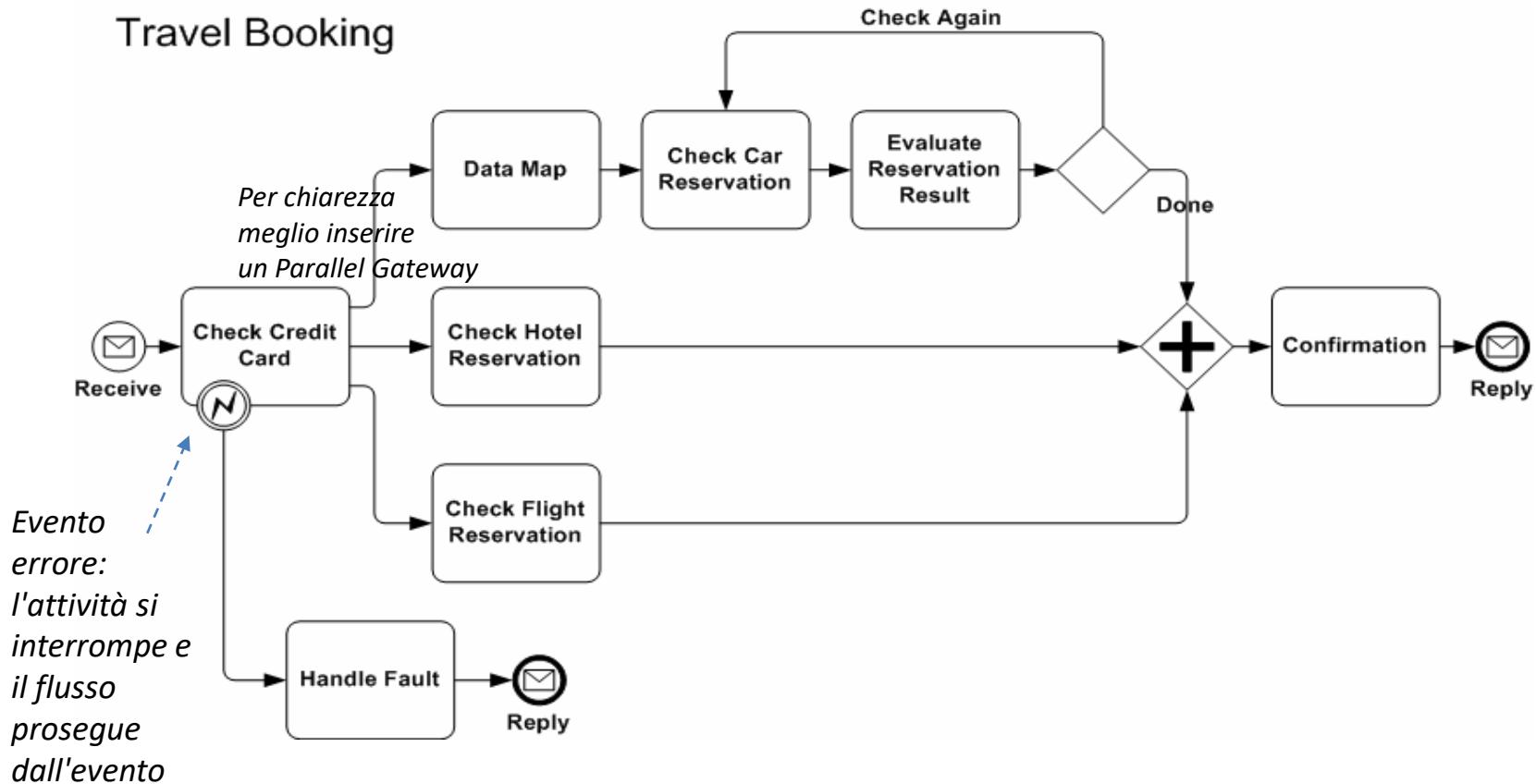
Collaboration



Il Pool non deve necessariamente contenere un processo

Il Pool Customer è una "scatola nera", dato che non si conosce il dettaglio del processo del cliente, ma solo le **interazioni (message flow)** con il Pool del processo Manufacturer

Esempio di modello BPMN



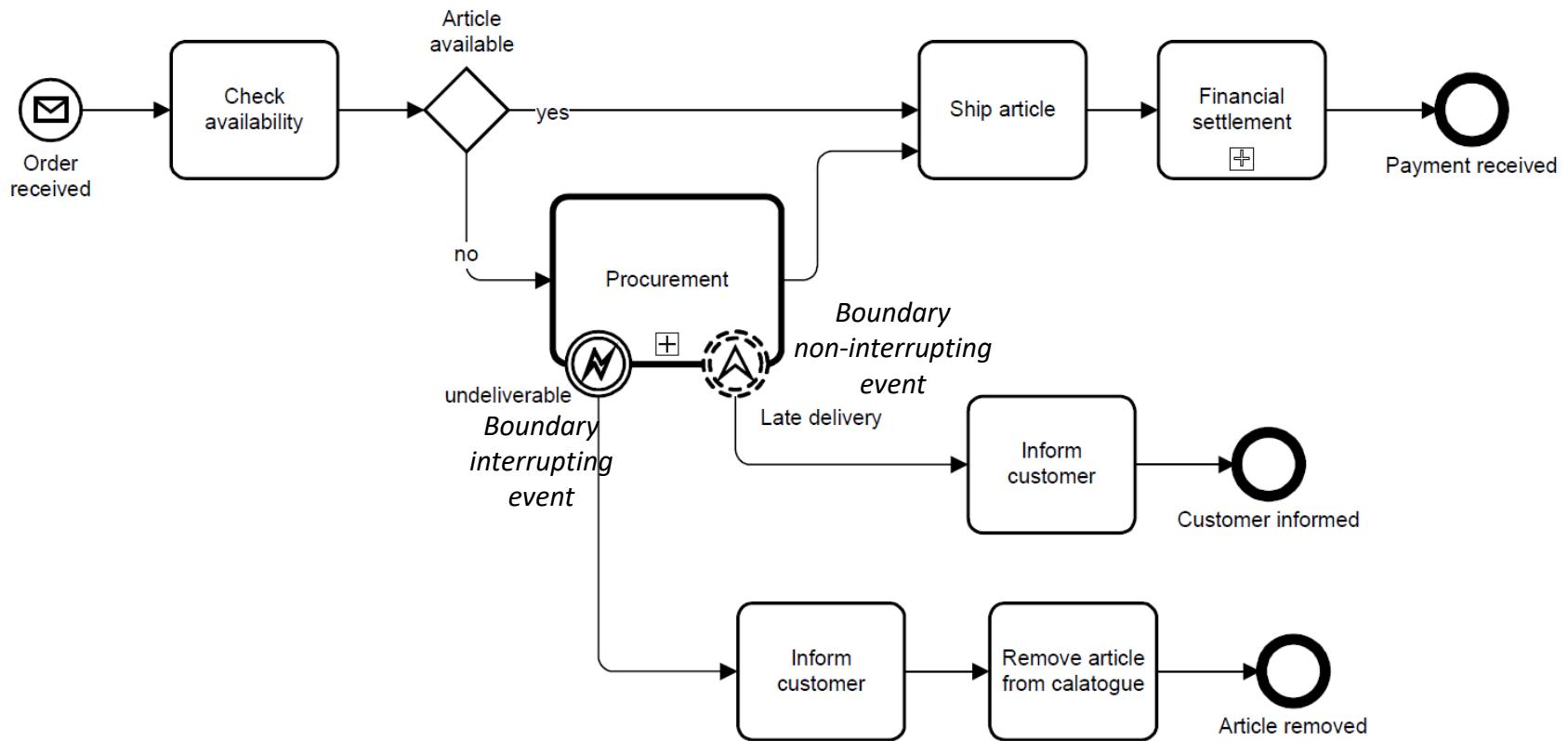
Caratteristiche di un buon modello

- **Saliente** : nessun modello può rappresentare ogni aspetto possibile, deve rappresentare esclusivamente ciò che più rilevante nel caso specifico
- **Accurato**: il modello dovrebbe codificare precisamente la situazione attuale e non fornire una vista errata o polarizzata
- **Completo ma parsimonioso**: il modello dovrebbe essere il più semplice possibile, ma non più semplice
- **Comprendibile** : il modello dovrebbe consentire a chi lo osserva di percepirlne il senso, e non dovrebbe essere troppo complicato o strano

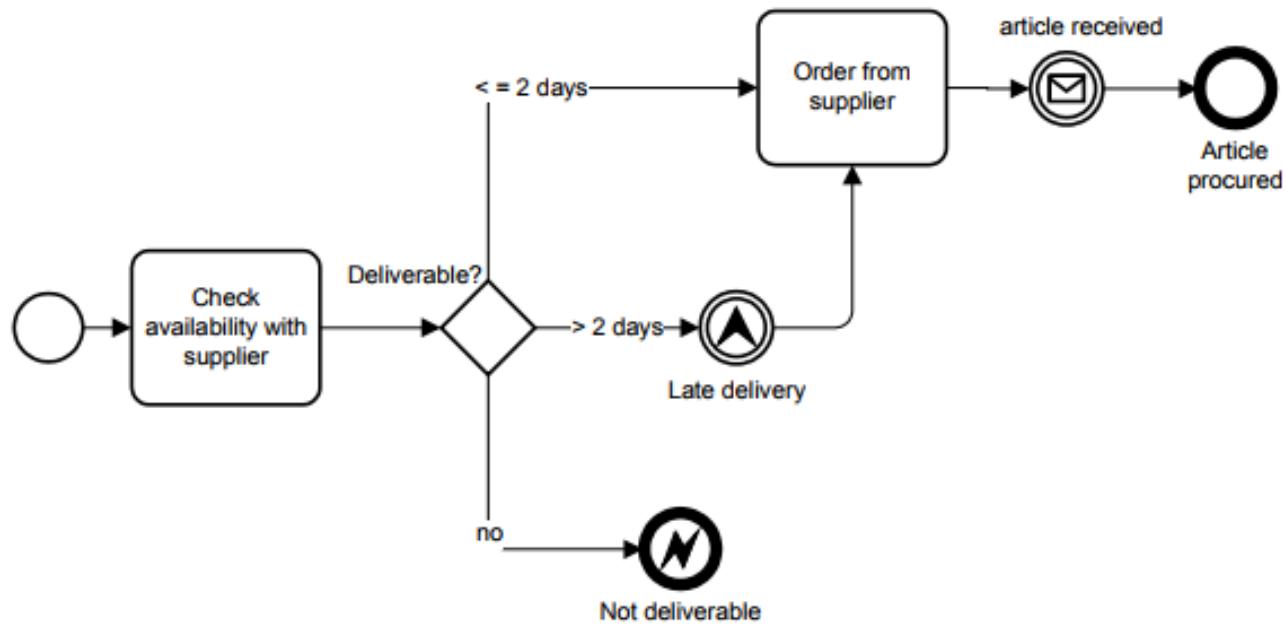
The order fulfillment process

1. The order fulfillment process must start after receiving an order message and continue to check whether the ordered article is available or not.
2. An available article is shipped to the customer followed by a financial settlement, which can be collapsed as a sub-process in this diagram.
3. In case that an article is not available, it has to be procured by calling the procurement subprocess (thickly bordered to mean a call activity). The procurement sub-process can fire two different events:
 1. An interrupting event in case the ordered article is not in the market anymore;
 2. An non-interrupting event in case the ordered article is not available but it can be resupplied
4. The customer must be informed about problems in the Procurement.
5. In case the ordered article cannot be resupplied, it must be removed from the Catalogue

The order fulfillment process



Il sottoprocesso di approvvigionamento



Offerta e gestione di mutui

Un rappresentante del servizio clienti invia un'offerta di mutuo ad un cliente e attende una risposta:

- se il cliente chiama o scrive per declinare l'offerta, la pratica viene aggiornata e archiviata;
- se il cliente invia i documenti dell'offerta compilati e allega tutti i documenti richiesti, la pratica viene trasferita all'amministrazione per completarla;
- se non sono stati forniti tutti i documenti richiesti viene inviata al cliente una richiesta di fornire i documenti mancanti;
- se dopo due settimane non viene ricevuta alcuna risposta, la pratica viene aggiornata e archiviata.

Offerta e gestione di mutui

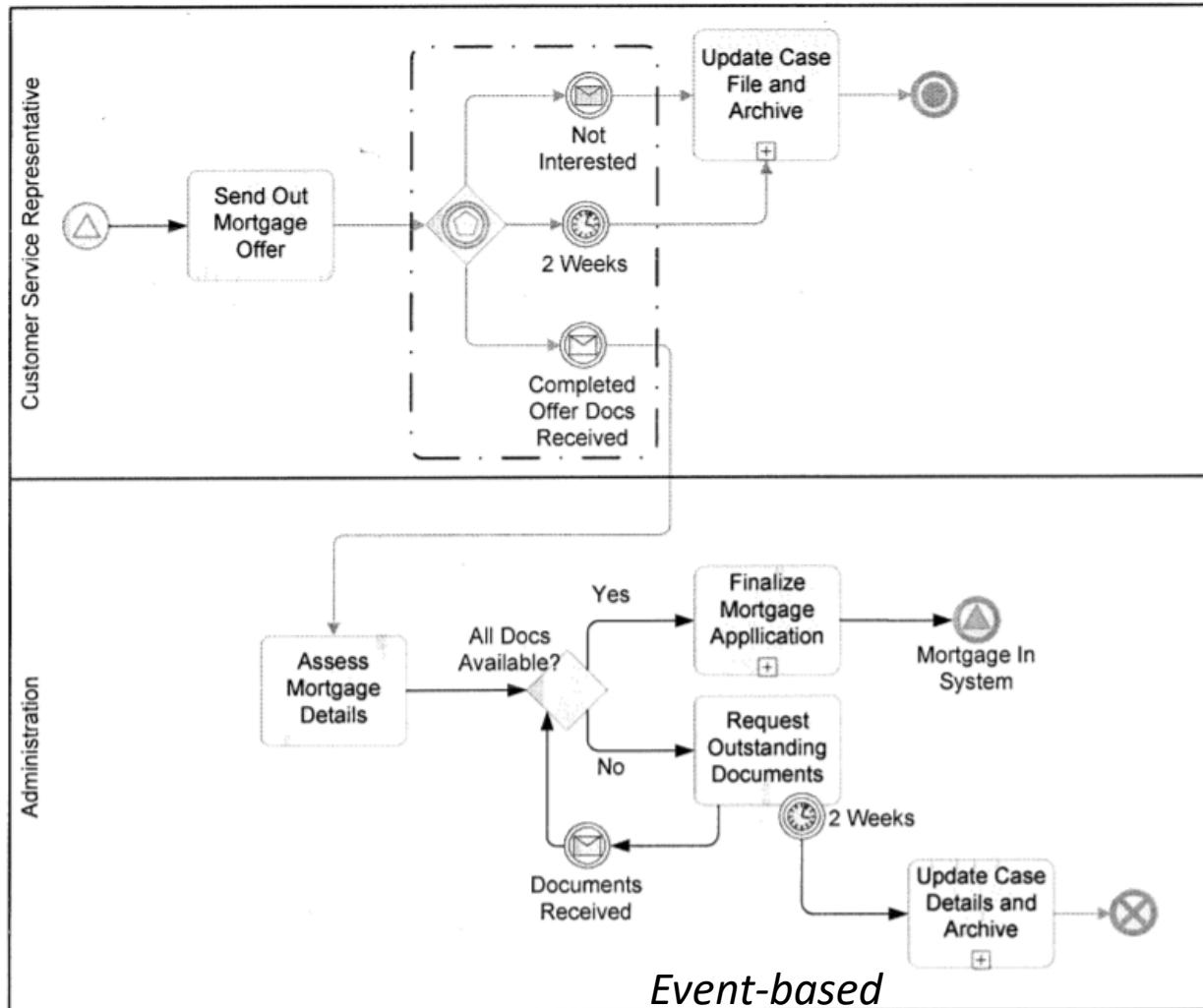
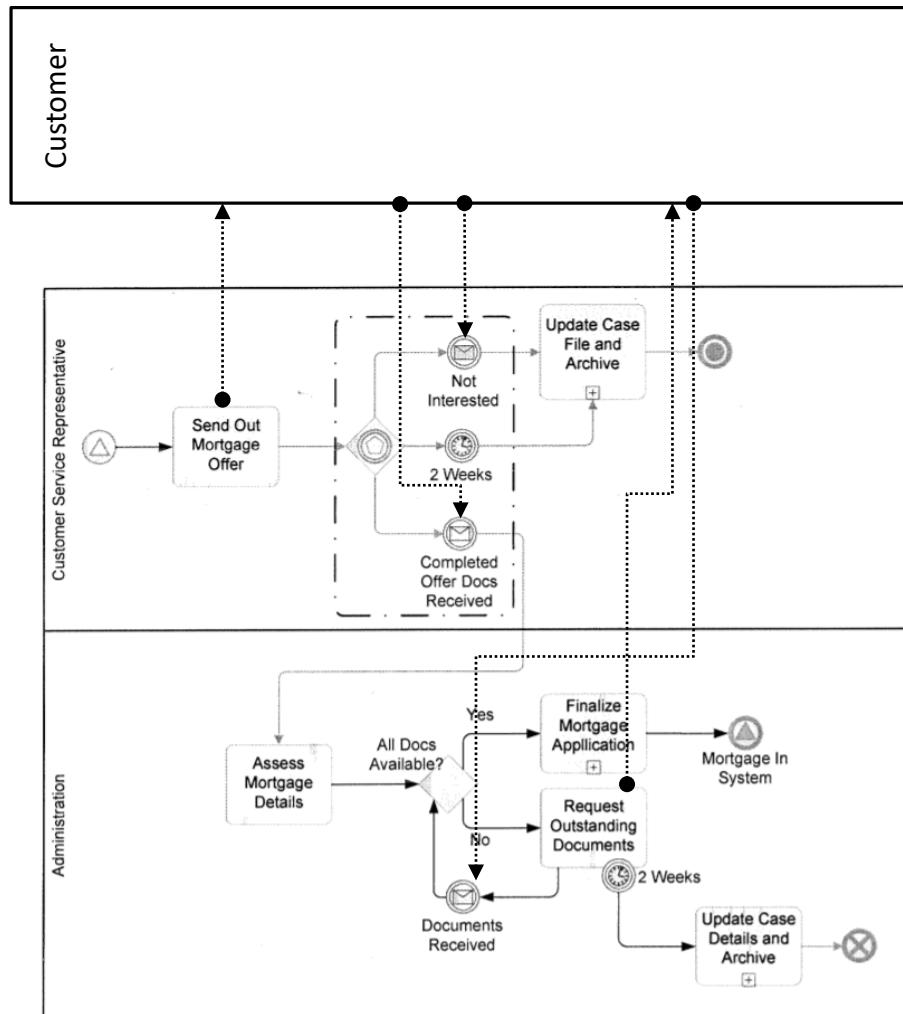


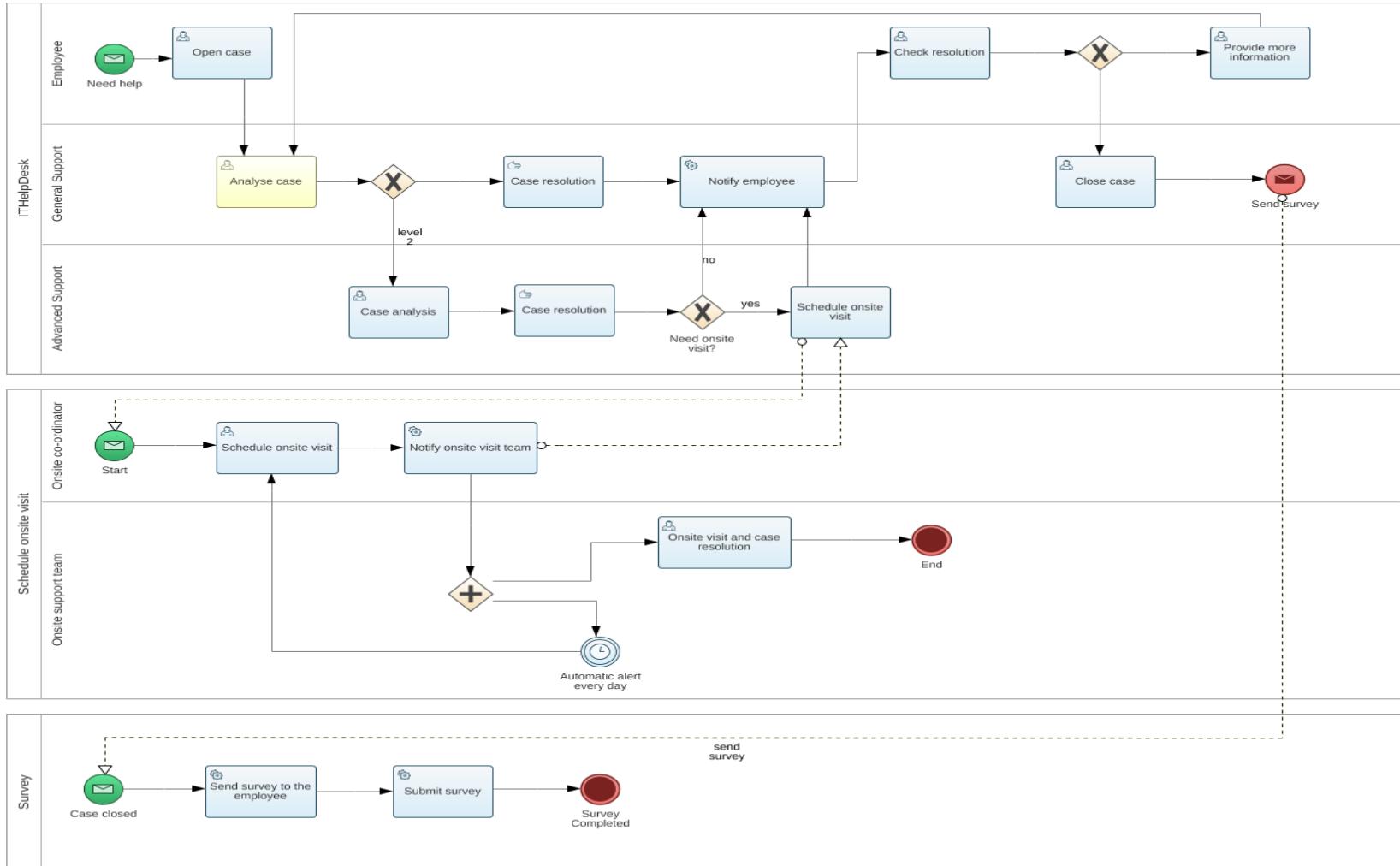
Diagramma
di processo

Offerta e gestione di mutui

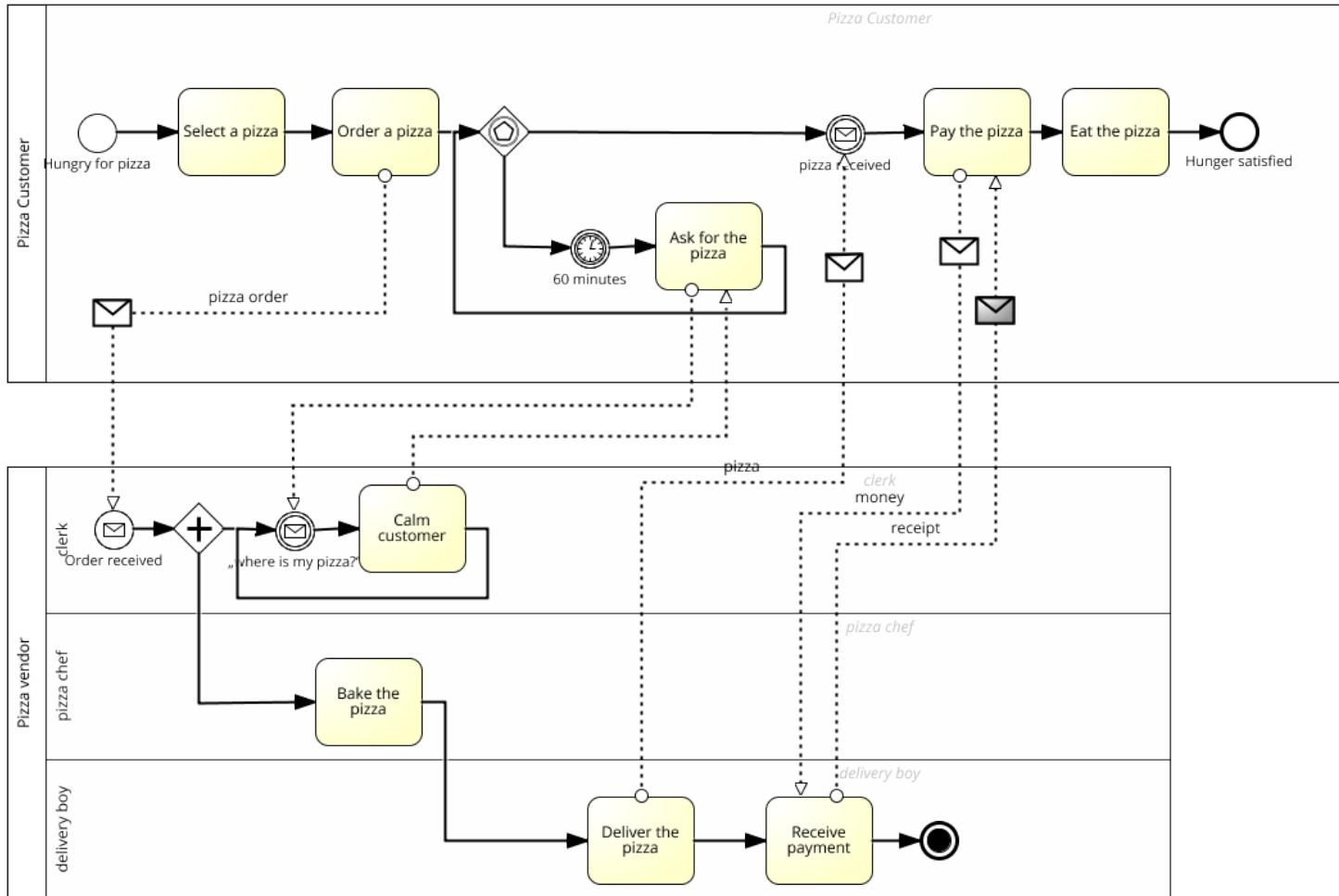


*Diagramma
di collaborazione*

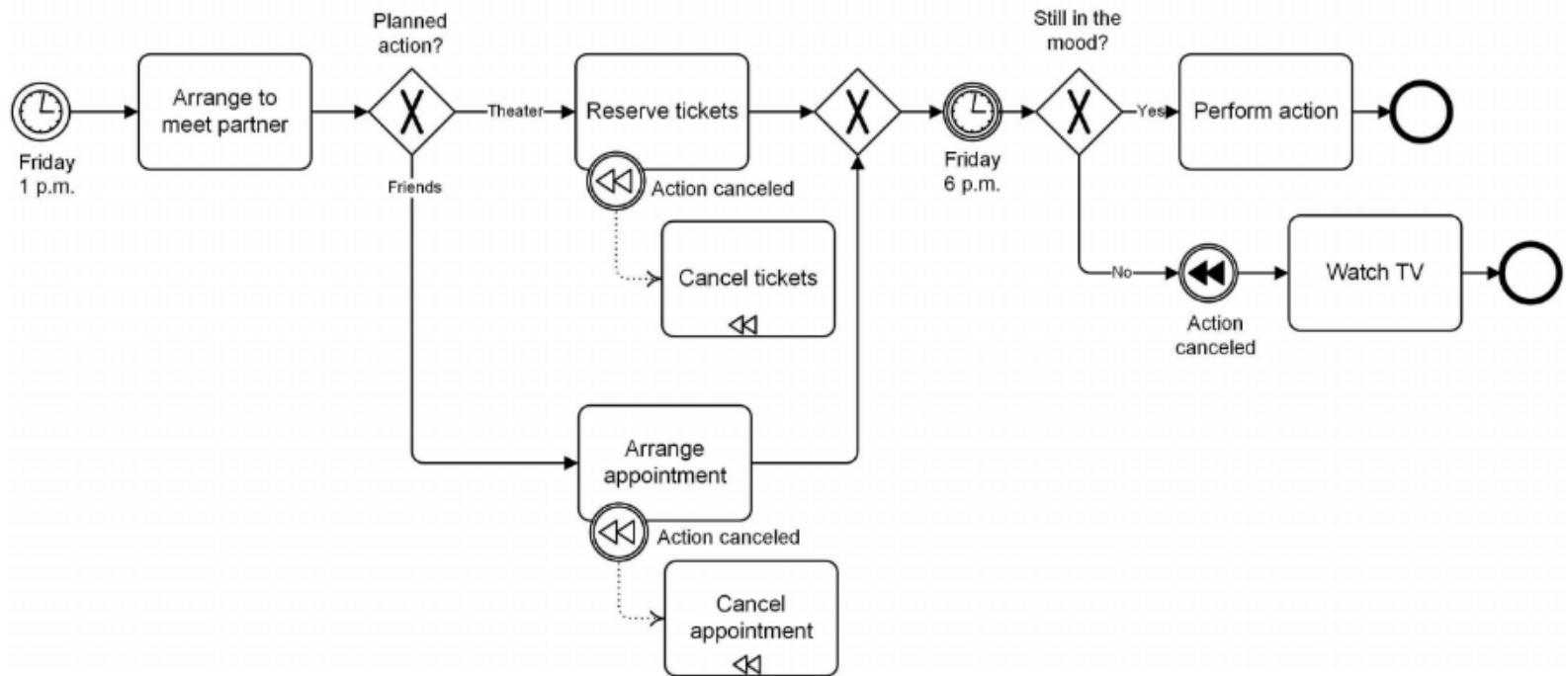
IT HelpDesk



Consegna della pizza



Compensazioni



Regole speciali

- l'evento è attivo solo nello stesso processo (pool)
- a differenza degli altri eventi attaccati (che hanno effetto se attività a cui sono attaccati rimangono attive), un evento di compensazione ha solo effetto se il processo lancia l'evento e l'attività che ha l'evento attaccato si era completata con successo
- l'evento attaccato è collegato al task di compensazione (un tipo speciale di task) tramite un'associazione (non un sequence flow)

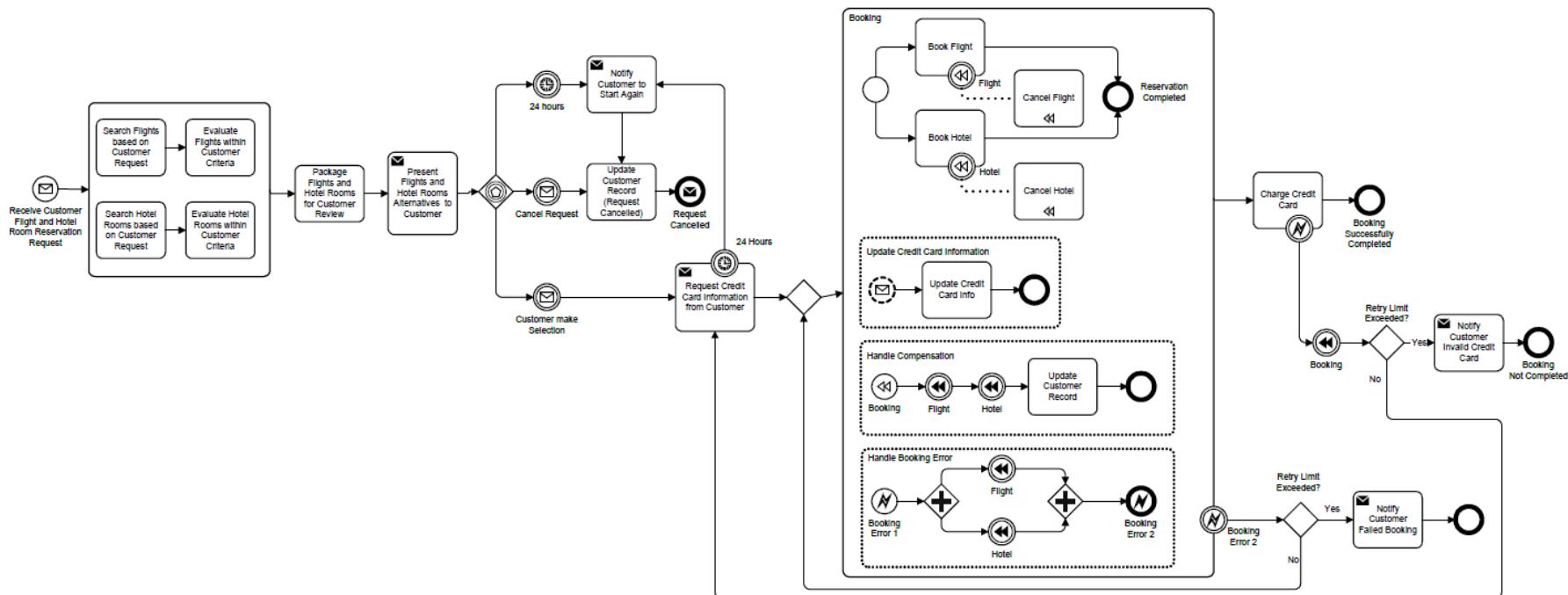
Prenotazione di viaggio

- L'agenzia di viaggi riceve una richiesta di prenotazione di viaggio, compresi il trasporto aereo e la prenotazione dell'hotel, da un Cliente.
- Dopo la ricerca e la valutazione della disponibilità dei voli e delle camere d'albergo, le alternative selezionate vengono confezionate e offerte al cliente.
- Il cliente ha 24 ore di tempo per scegliere un'alternativa proposta o per cancellare la richiesta. In caso di cancellazione, o dopo lo scadere del termine, l'Agenzia aggiorna il record del cliente per riflettere la cancellazione della richiesta e il cliente viene avvisato.
- Quando viene effettuata una selezione, al Cliente viene chiesto di fornire le informazioni della carta di credito. Anche in questo caso, il Cliente ha 24 ore per fornire queste informazioni o la richiesta viene annullata tramite le stesse attività indicate prima (aggiornamento e notifica).

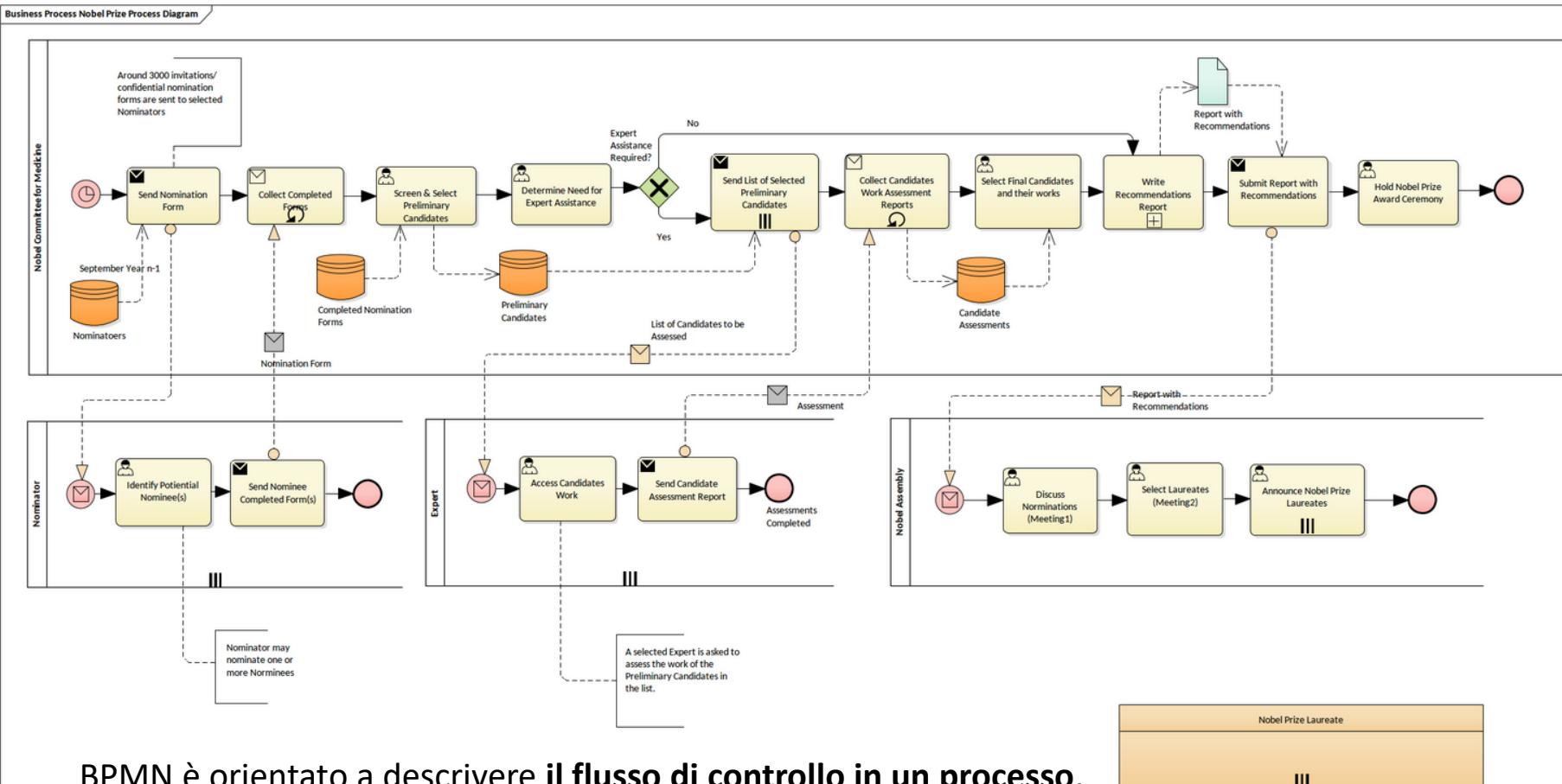
Dopo aver ricevuto i dati della carta di credito, le attività di prenotazione hanno luogo:

- il volo e la camera d'albergo vengono prenotati. Vengono prese misure per assicurare l'annullamento delle prenotazioni in caso di problemi nelle attività di prenotazione e pagamento. Il Cliente ha anche il diritto di fornire all'Agenzia modifiche ai dati della carta di credito prima che la prenotazione sia completata. Tali informazioni saranno salvate nel suo record;
- se si verifica un errore durante le attività di prenotazione, le prenotazioni di voli e camere d'albergo vengono annullate e il record del Cliente viene aggiornato. La prenotazione viene ritentata fino a quando non viene superato il limite di tentativi della prenotazione;
- se la prenotazione va a buon fine, le prenotazioni vengono addebitate sulla carta di credito del cliente e il processo si ferma dopo la conferma di successo. Se si verifica un errore durante questa attività, la prenotazione del volo e della camera d'albergo viene annullata. Al cliente viene chiesto di nuovo di fornire i dati della carta di credito e la prenotazione viene ritentata fino a quando non viene superato il limite di ripetizione dell'elaborazione del pagamento;
- in entrambi i casi, dopo l'errore, quando il limite di tentativi viene superato, il Cliente viene avvisato e il processo si ferma.

Prenotazione di viaggio



Uso di Data Object e Data Store



BPMN è orientato a descrivere **il flusso di controllo in un processo**, non il flusso dei dati che possono essere comunque rappresentati tramite associazioni e artefatti (Data Object e Data Store)

Introduzione ai Data Flow Diagram

Analisi funzionale

- modellazione di un sistema informativo in termini di
 - attività o processi e
 - flussi informativi tra essi
- genera lo *schema funzionale*
 - descrive il trattamento dell'informazione nella base di dati
 - complementare alla descrizione strutturale fornita dallo *schema concettuale* (E/R)

Analisi funzionale (ii)

- si può concentrare su diversi aspetti
 - dati e flussi scambiati tra attività
 - adatti a sistemi tradizionali di elaborazione dati
 - sincronizzazione di attività tramite pre- e post-condizioni
 - adatti ad attività fortemente legate all’evoluzione temporale (sistemi *real time*)

Data Flow Diagram

- Traggono origine dalla teoria dei grafi e sono precedenti all'avvento dei computer per la gestione delle informazioni
- Sono stati impiegati per la prima volta, nel settore dell'ingegneria del software, solo verso la metà degli anni 70
- Non esiste in letteratura a tutt'oggi una definizione universalmente accettata e sono presenti molteplici differenti formulazioni operative

Che cosa modella un DFD?

- Un sistema è visto come **una rete di processi funzionali interconnessi da depositi di dati**
- I DFD enfatizzano le operazioni effettuate sulle informazioni e le dipendenze funzionali che vengono a crearsi fra i vari processi in base ai flussi di informazione
- I processi possono essere definiti a qualunque livello di astrazione, raffinabili mediante **scomposizione gerarchica** in un insieme di processi più elementari.

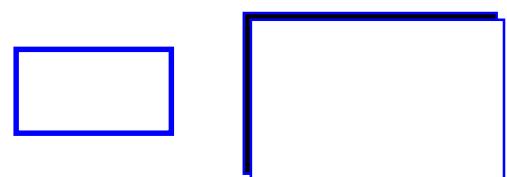
Concetti rappresentati in un DFD

- **processi** (detti anche bolle)
 - generano, usano o distruggono dati
 - trasformano dati di un *flusso di ingresso* in un *flusso di uscita*;
- **flussi** 
 - scambi di informazioni tra processi
 - **non** rappresentano i *controlli*



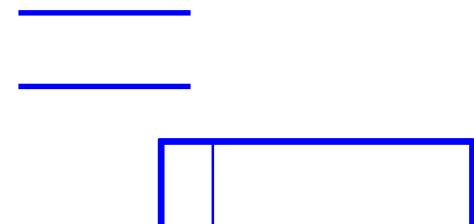
Concetti rappresentati in un DFD (ii)

- **agenti esterni** (detti anche *terminatori*, *interfacce* o *external entities*)
 - producono e/o consumano dati;



- **depositi di dati**
 - memorizzano informazioni in modo passivo

- archivi temporanei o permanenti
 - tavole di consultazione
 - moduli cartacei o elettronici



- una linea da un processo a un deposito significa che il processo modifica i dati
 - una linea da un deposito a un processo significa che il processo usa i dati

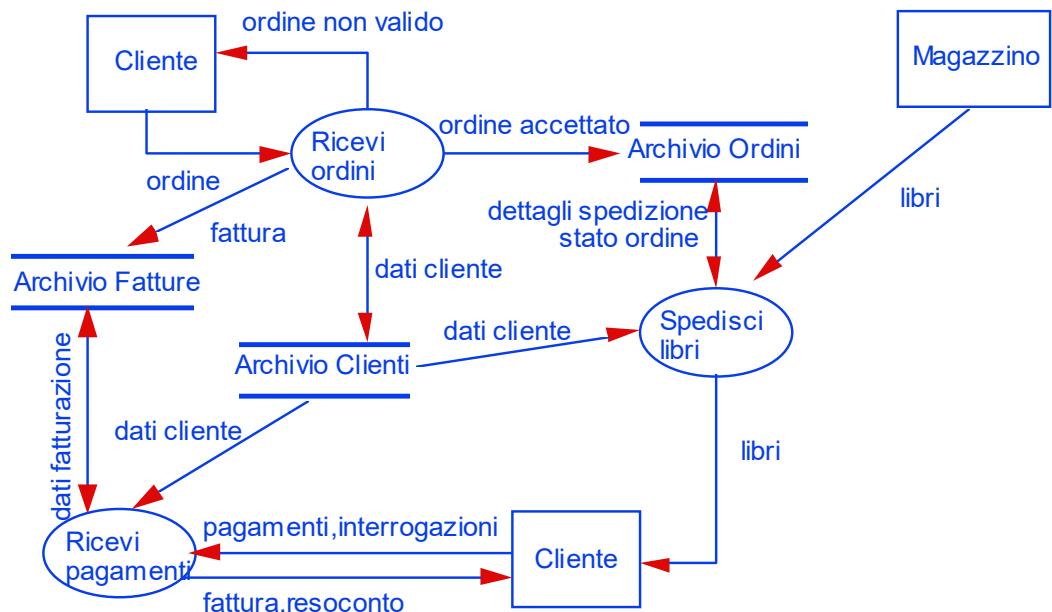
DFD : sintassi

- un DFD è una quadrupla $\langle P, D, A, F \rangle$ dove:
 - $P = \{p_1, p_2, \dots, p_n\}$ è un insieme finito, non vuoto, di processi;
 - $D = \{d_1, d_2, \dots, d_r\}$ è un insieme finito di depositi;
 - $A = \{a_1, a_2, \dots, a_s\}$ è un insieme finito di agenti;
 - $F = \{ f \in (P \times (P \cup D \cup A)) \cup ((P \cup D \cup A) \times P) \}$ è un insieme finito di flussi.

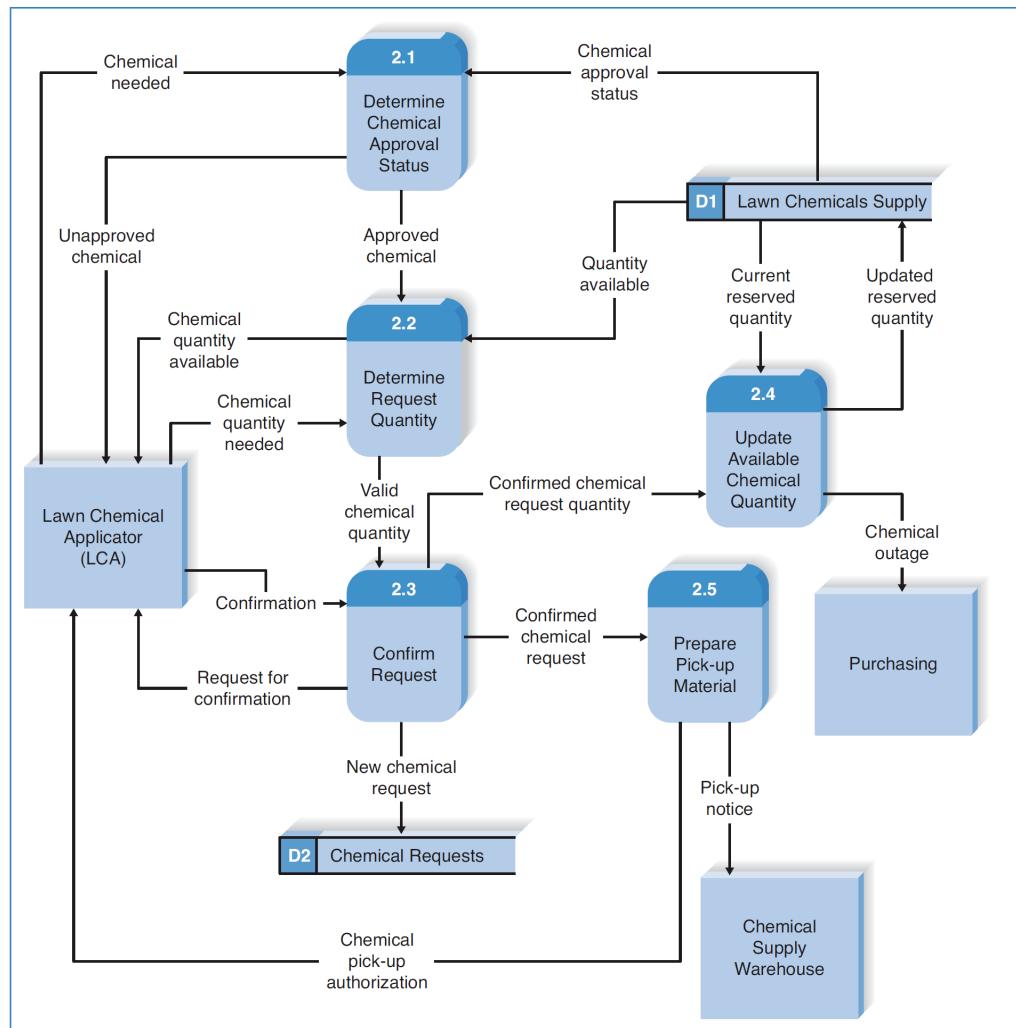
Rappresentazione a Grafo

- grafo orientato in cui ogni nodo appartiene a uno dei tre insiemi P,D o A, e ogni arco orientato rappresenta un flusso di dati

Un DFD che descrive i rapporti fra una libreria e i suoi clienti



Esempio



Consigli

- Nomi univoci per identificare processi, flussi di dati, agenti e depositi
- Non rappresentare le eccezioni e il trattamento degli errori, rimandando questi dettagli alle fasi finali dell'analisi
- Un DFD non è un diagramma di flusso, dove le frecce indicano un ordinamento negli eventi
- In alcune estensioni, come nella notazione OMT è possibile rappresentare anche flussi di controllo; anticiparli in un DFD comporta di fatto una duplicazione se si usa anche un modello dinamico

Regole da rispettare

- Scegliere nomi significativi per i processi, flussi, depositi, e agenti
- Numerare i processi
- Disegnare i DFD seguendo criteri estetici
- Evitare DFD eccessivamente complessi
- Accertarsi della coerenza interna di un DFD e che sia coerente con quelli ad esso associati

Vincoli

- Evitare pozzi infiniti
- Sospettare di processi a generazione spontanea
- Depositi a sola lettura o a sola scrittura sono rari
- Non devono esistere flussi di dati fra :
 - due agenti esterni
 - due depositi
 - un'entità esterna e un deposito

Avvertenze

- L'assenza di nomi per flussi, processi e depositi può essere indice di trascuratezza, di indecisione ma può nascondere anche insidie più gravi, ad esempio che l'analista confonde tra un flow-chart e un DFD
- C'è una convenzione, spesso accettata in molte versioni dei DFD, che un flusso da o verso un deposito possa essere non etichettato quando i dati trasferiti corrispondono ad un oggetto (record) intero



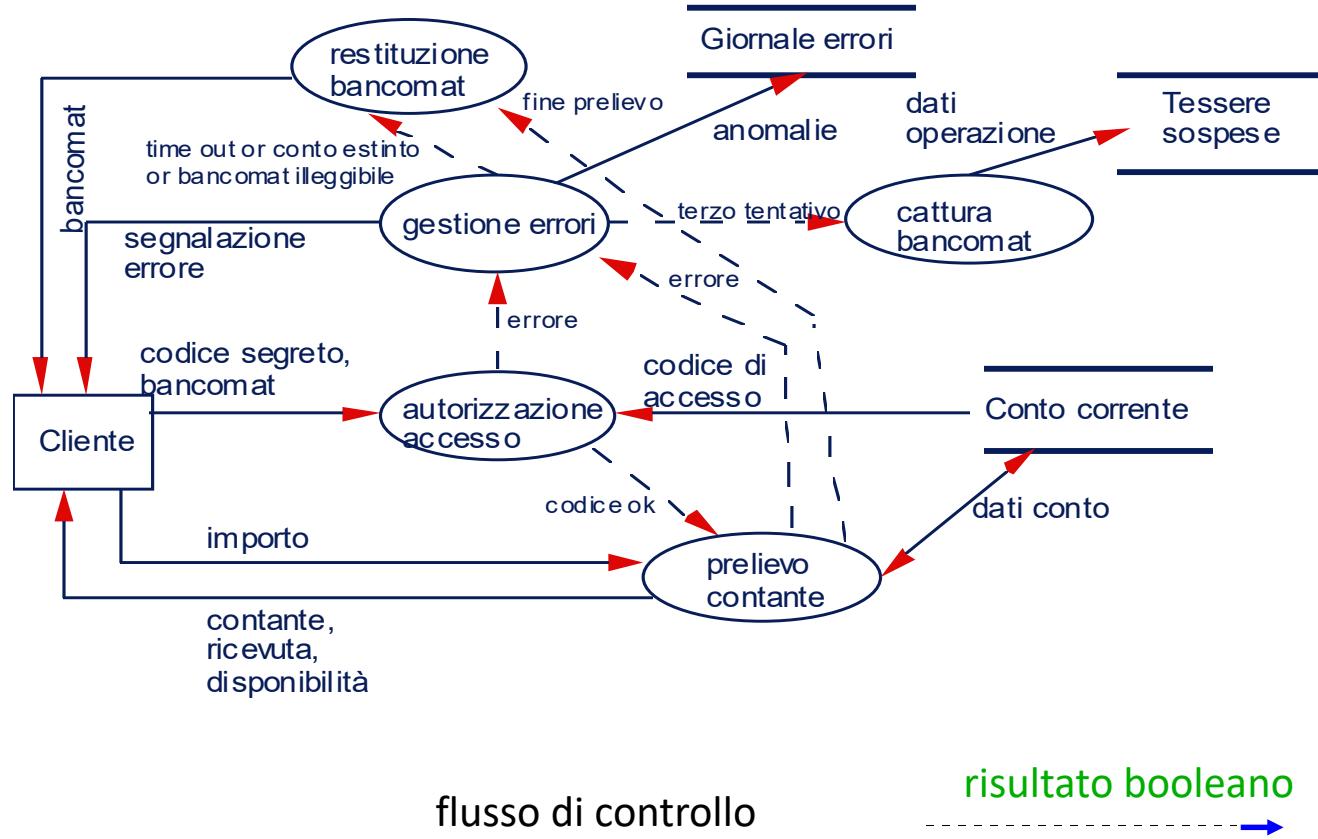
DFD estesi

- Alcune notazioni fanno uso di una più ricca simbologia grafica per l'esplicitazione di alcuni aspetti che di per sé un DFD non mostra immediatamente



*Il conto corrente è visto
sia come un oggetto da memorizzare nell'archivio della banca,
sia come un deposito dati relativo ad un cliente da utilizzare per vari tipi di operazioni*

DFD con flussi di controllo



Diagrammi UML

- **Activity Diagram**
- **Use Case Diagram**
- **Class Diagram**
- **Sequence Diagram**
- **Collaboration Diagram**
- **Statechart Diagram**



Activity Diagram (diagrammi d'attività)

- Rappresentano una procedura o un workflow
- Mostrando l'evoluzione di un flusso di attività
- Ogni attività è definita come un'evoluzione continua, non necessariamente atomica, di uno stato
- Sono una evoluzione dei flow-chart

Activity Diagram: elementi base

- Activity (Attività)

Esecuzione non atomica entro un sistema dotato di stati.

Può essere scomposta in azioni.

- Action (Azione)

Operazione atomica eseguibile che produce come risultato un cambiamento nello stato di un sistema o il ritorno di un valore.

Activity Diagram: concetti base

- Action State (Stato di azione)

Uno stato che rappresenta l'esecuzione di un'azione (atomica), tipicamente l'invocazione di una operazione.

- Activity State (Stato di attività)

Stato composito, in cui il flusso di controllo è formato di altri stati di attività e stati di azione. Non è atomico, il che significa anche che può essere interrotto. Può anche essere ulteriormente scomposto in altri diagrammi di attività

Activity Diagram: elementi di contorno

- Transition (Transizione)

Rappresenta il flusso di controllo fra due attività, che mostra il percorso da un action o activity state al successivo action o activity state.

- Object Flow

Rappresenta un oggetto (un'entità) coinvolta nel flusso di controllo associato con un activity diagram.

Activity Diagram: elementi di contorno

- Object State

Una condizione o situazione operativa nella vita di un oggetto (un'entità) durante la quale l'oggetto soddisfa certe condizioni, compie certe attività o attende certi eventi.

- Swimlane

Una suddivisione per l'organizzazione di responsabilità per le attività. Non ha un significato fisso, ma spesso corrisponde alla

Activity Diagram: tipologie

- Activity state (Diagram)

Le singole attività hanno una durata e possono essere ulteriormente scomposte, dando origine ad altri diagrammi

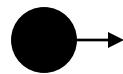
- Action state (Diagram)

Le singole attività sono atomiche e non possono essere ulteriormente scomposte

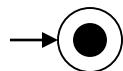
Activity Diagram: significato

- Enfasi posta sulle attività e non su chi le compie
- Enfasi sulla sequenza di azioni di una particolare procedura
- Vengono evidenziati vincoli di precedenza o di concorrenza

Activity Diagram: simboli base



Inizio di un diagramma di attività



Termine di un diagramma di attività

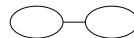
Attività 1

Singola attività generica con nome “Attività 1”



Connessione fra attività

Attività 2



Subattività generica (scomponibile)

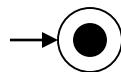
Non vi è differenza fra simboli di attività e simboli di azione

Nome del diagramma di attività (in fondo)

Activity Diagram: simboli base



Inizio di un diagramma di attività



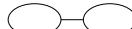
Termine di un diagramma di attività

Attività 1

Singola attività o azione generica
con nome “Attività 1”



Attività 2



Connessione fra attività

Attività generica di cui viene
esplicitata la scomponibilità

Processo 1

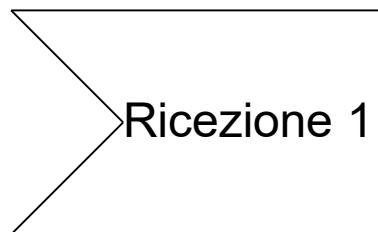
Processo generico con nome “processo 1”

*Nuove icone per definire attività/processi di business
nell'estensioni UML for Business di Eriksson-Penker
(descritte più avanti)*

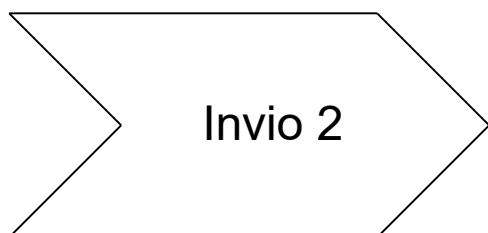
Activity Diagram: simboli di invio/ricezione



Attività con invio
con nome “Invio 1”

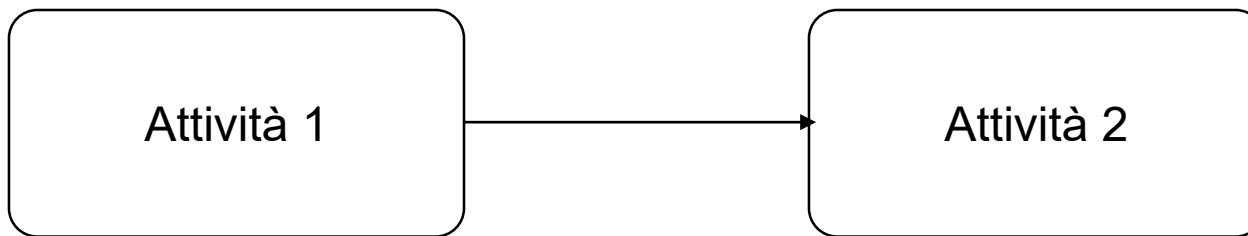


Attività con ricezione
con nome “Ricezione 1”



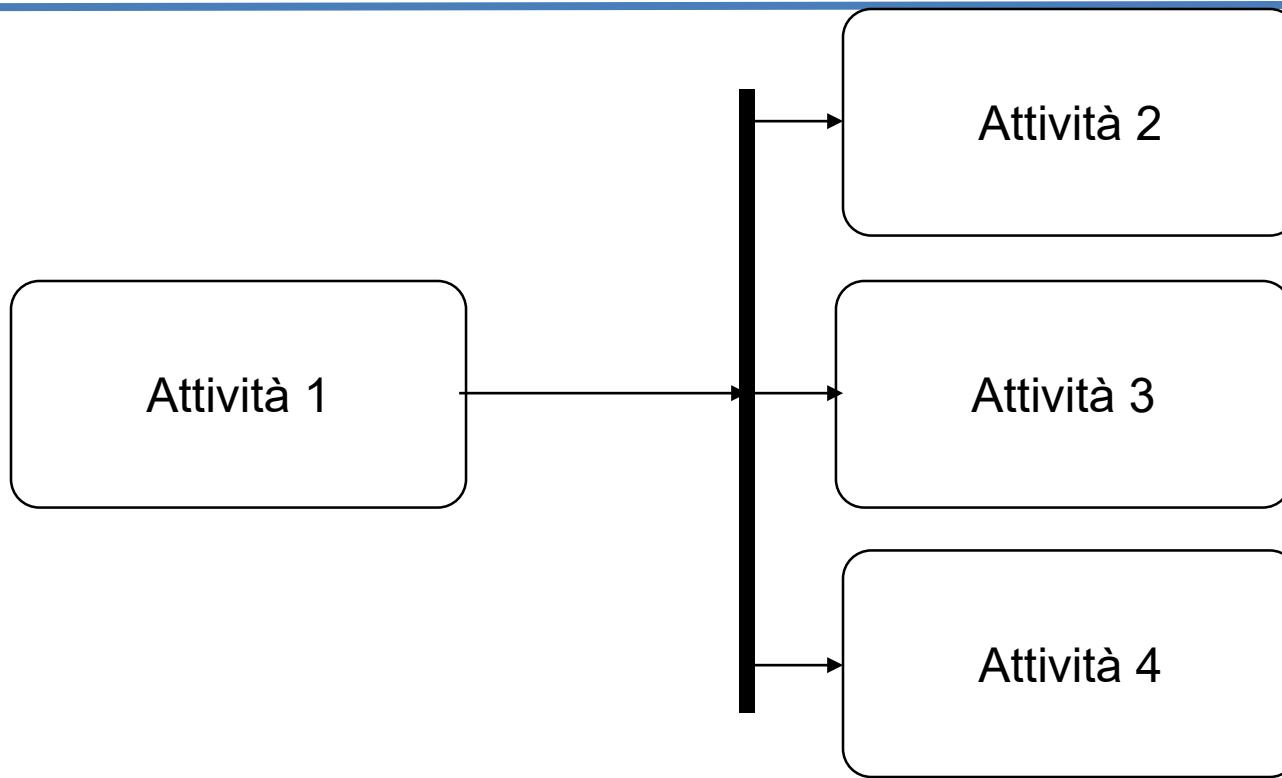
Attività con invio e ricezione
con nome “Invio 2”

Tipi di AD: sequenza semplice



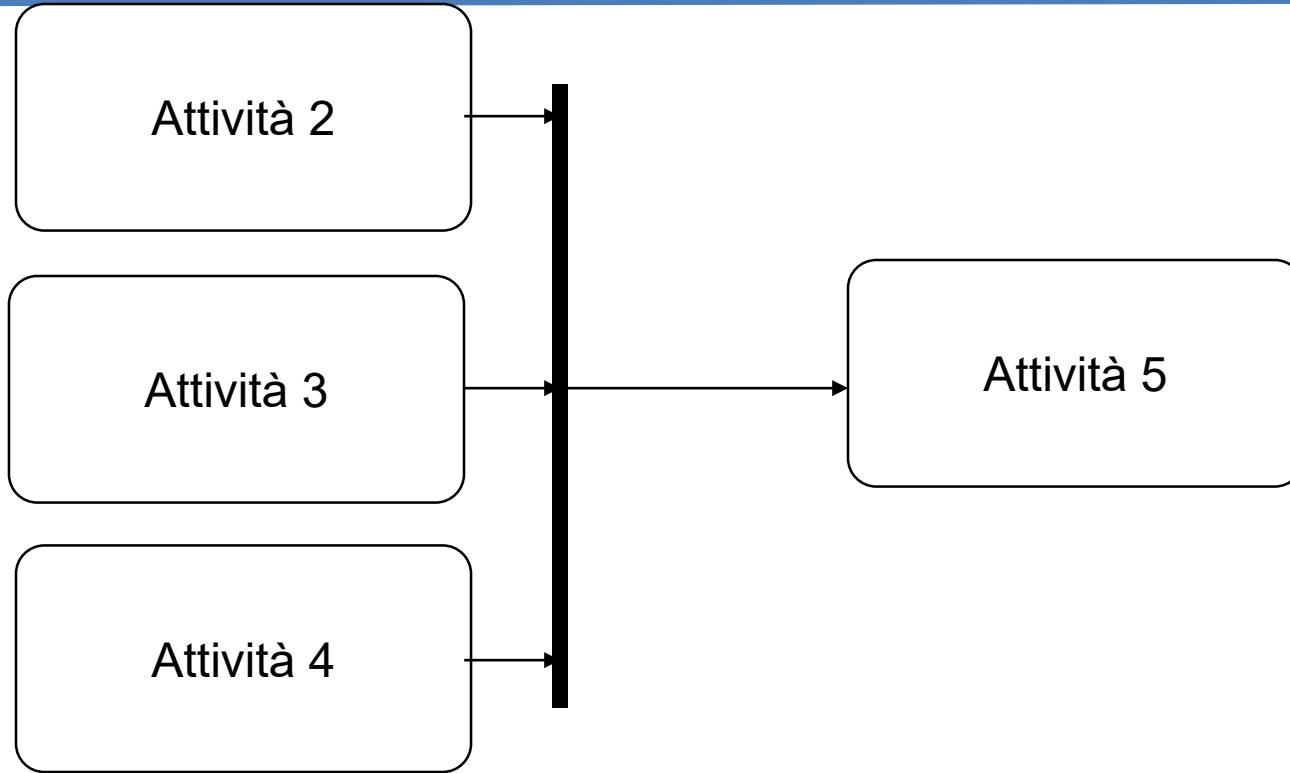
Un'attività (Attività 2) viene eseguita dopo la fine della
Precedente (Attività 1)
(Single Thread)

Tipi di AD: AND-split



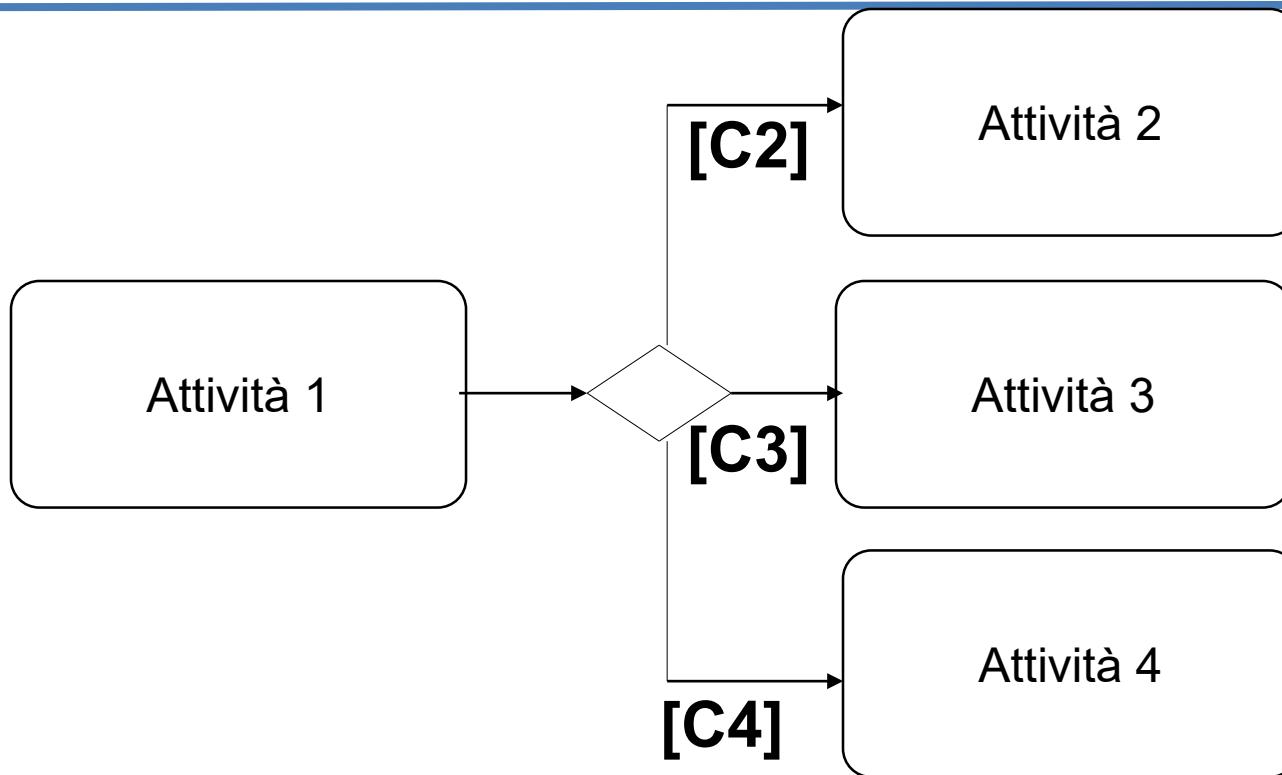
Un singolo flusso di attività si divide in più flussi, consentendo l'esecuzione simultanea di più attività
(Multiple Thread)

Tipi di AD: AND-join



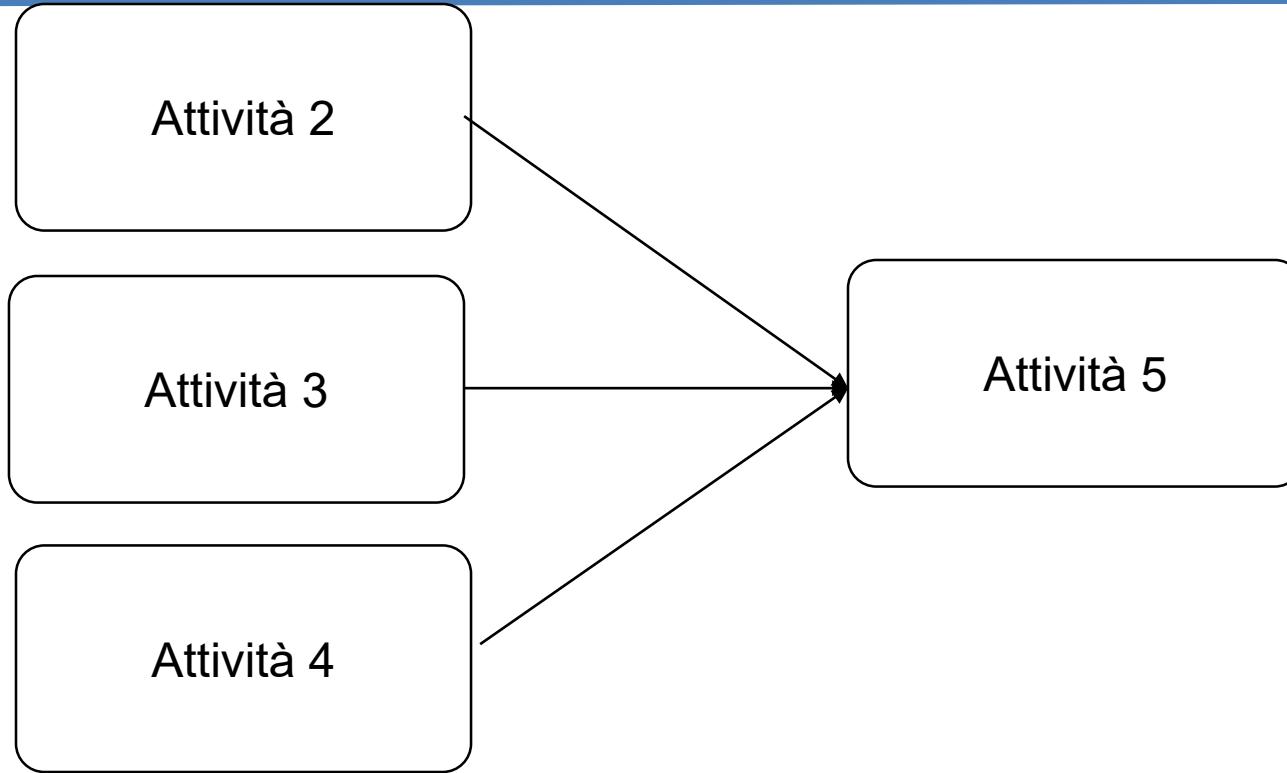
Due o più flussi di attività convergono in uno solo
E' un punto di sincronizzazione per il workflow: non si va
avanti finché non sono terminate tutte le attività precedenti

Tipi di AD: OR-split



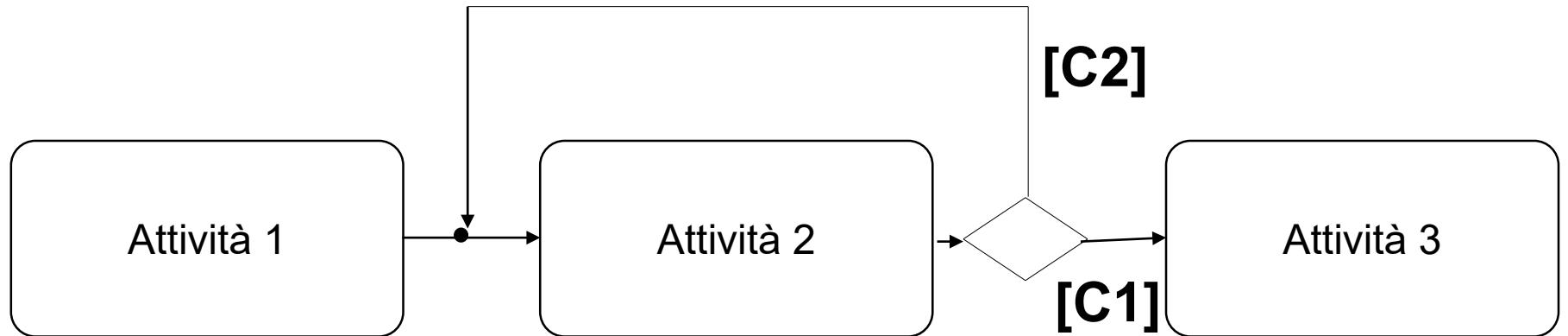
Un singolo flusso di attività prosegue per uno dei cammini in base al verificarsi delle condizioni di transizione, indicate fra parentesi quadre

Tipi di AD: OR-join



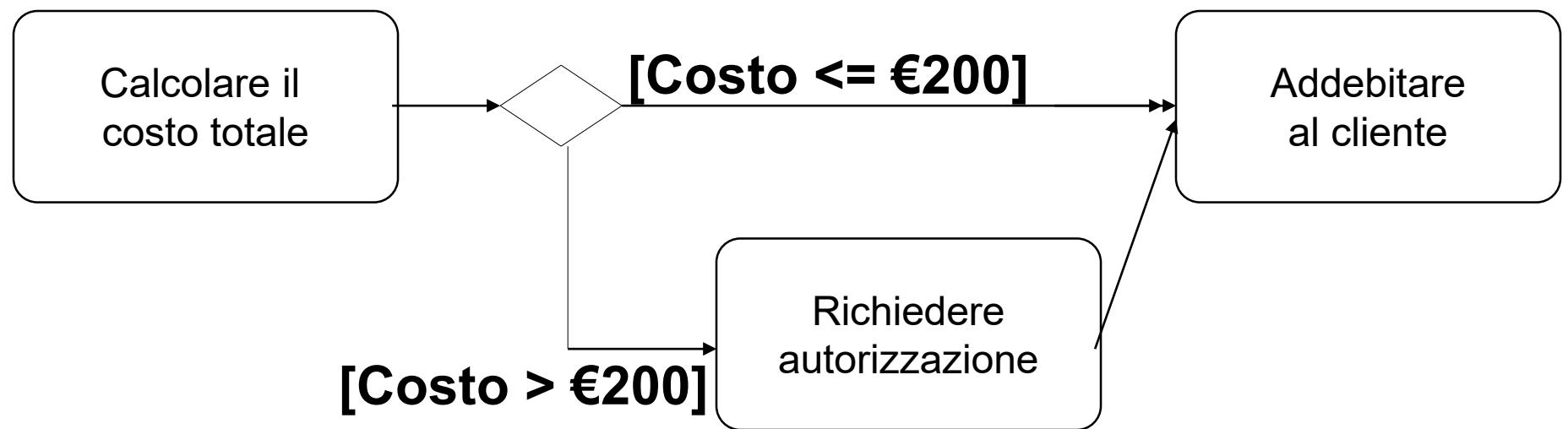
Un punto dove due o più flussi di attività ri-convergono in uno solo
ovvero hanno tutti Attività 5 come elemento successivo

Tipi di AD: iterazione



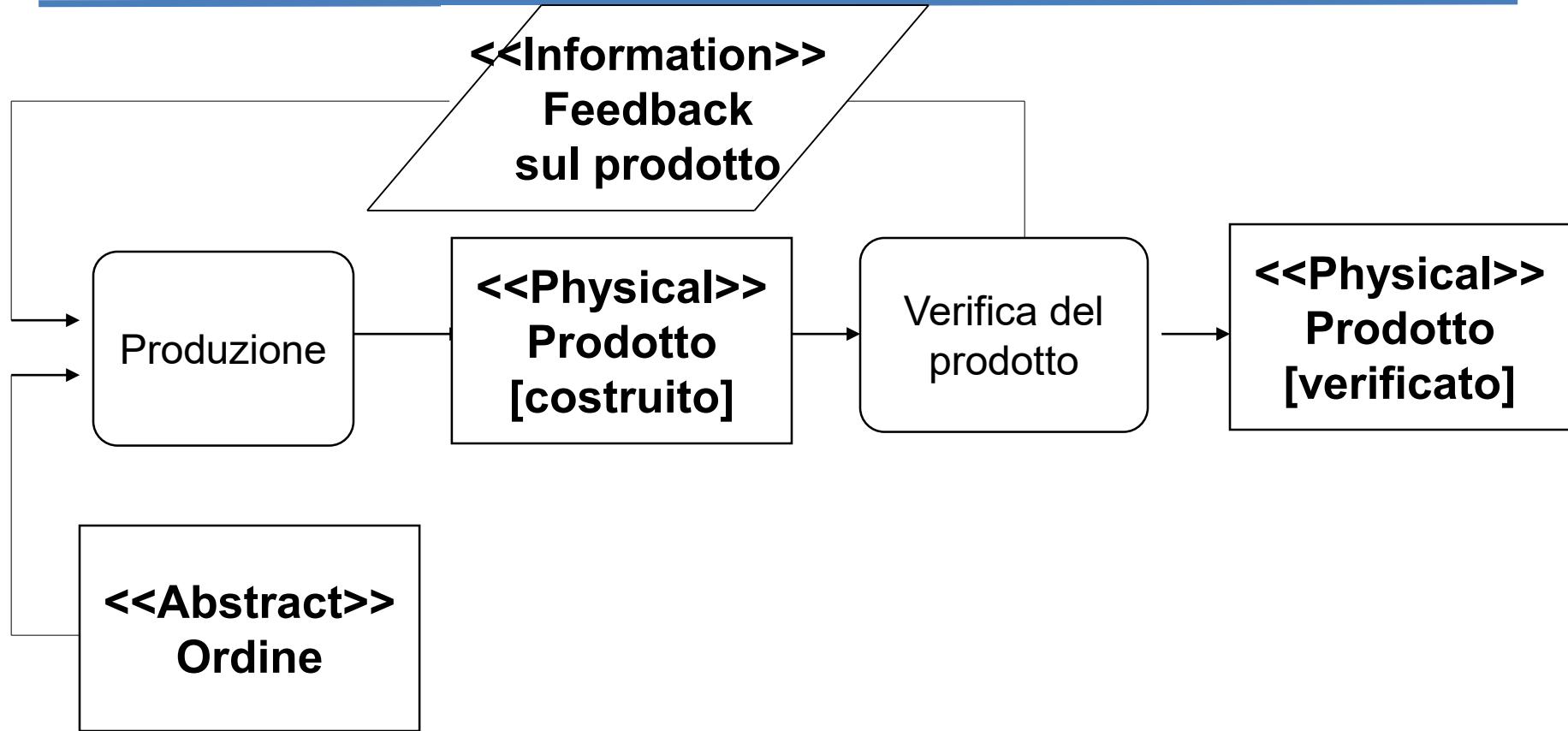
Un'attività (Attività 2) viene ripetuta più volte,
in base al verificarsi o meno di opportune condizioni
di controllo

Activity Diagram: esempi vari

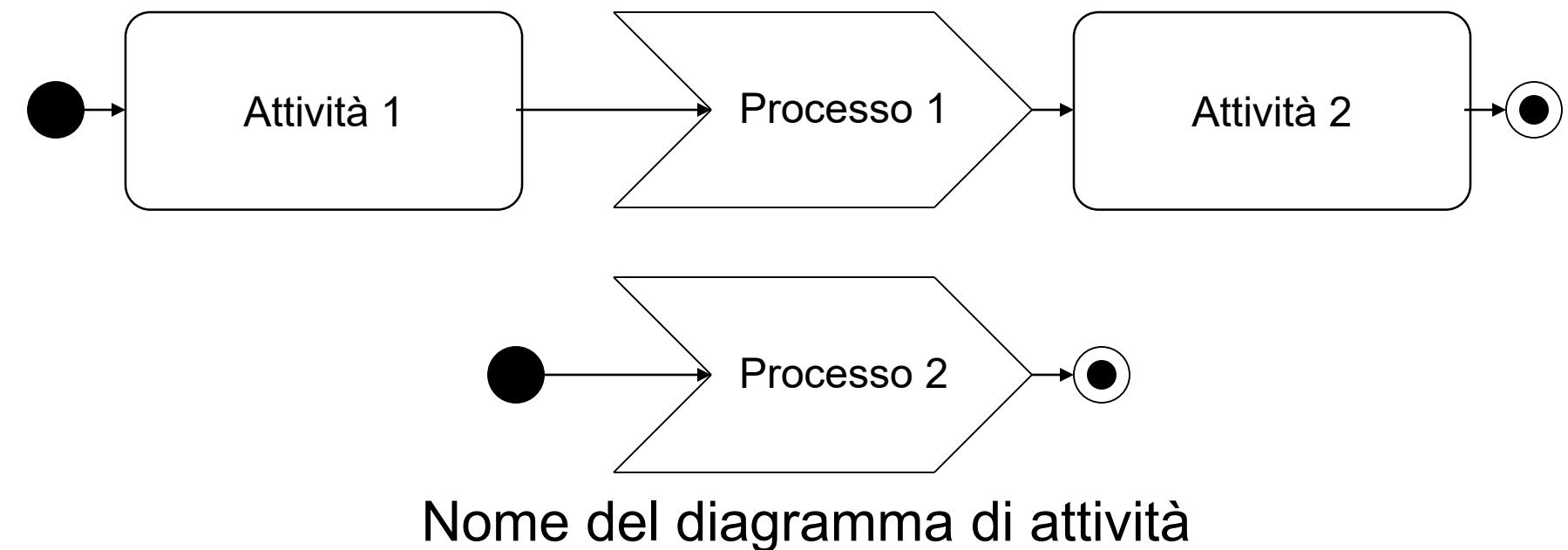


Se il costo totale è maggiore di 200€, bisogna chiedere l'autorizzazione prima di addebitarlo al cliente.

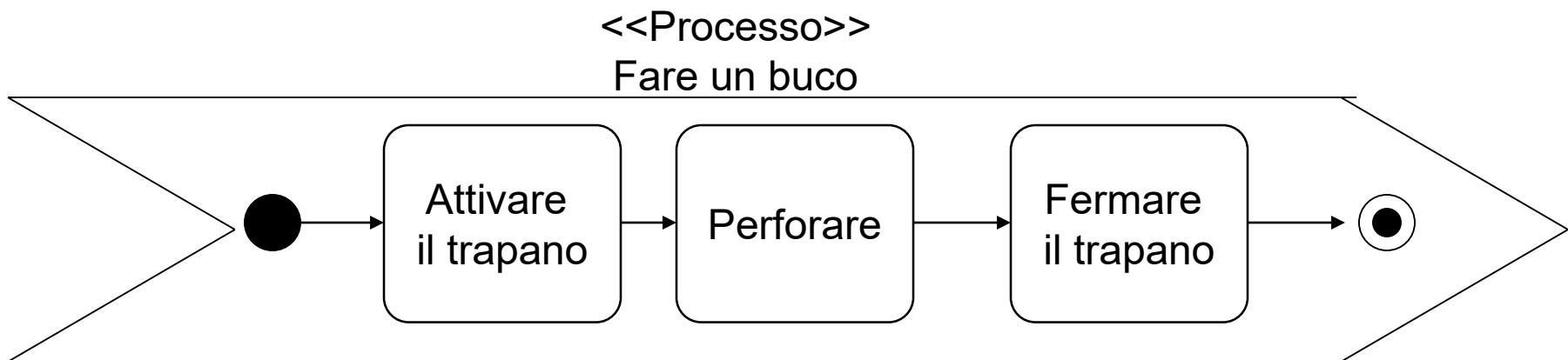
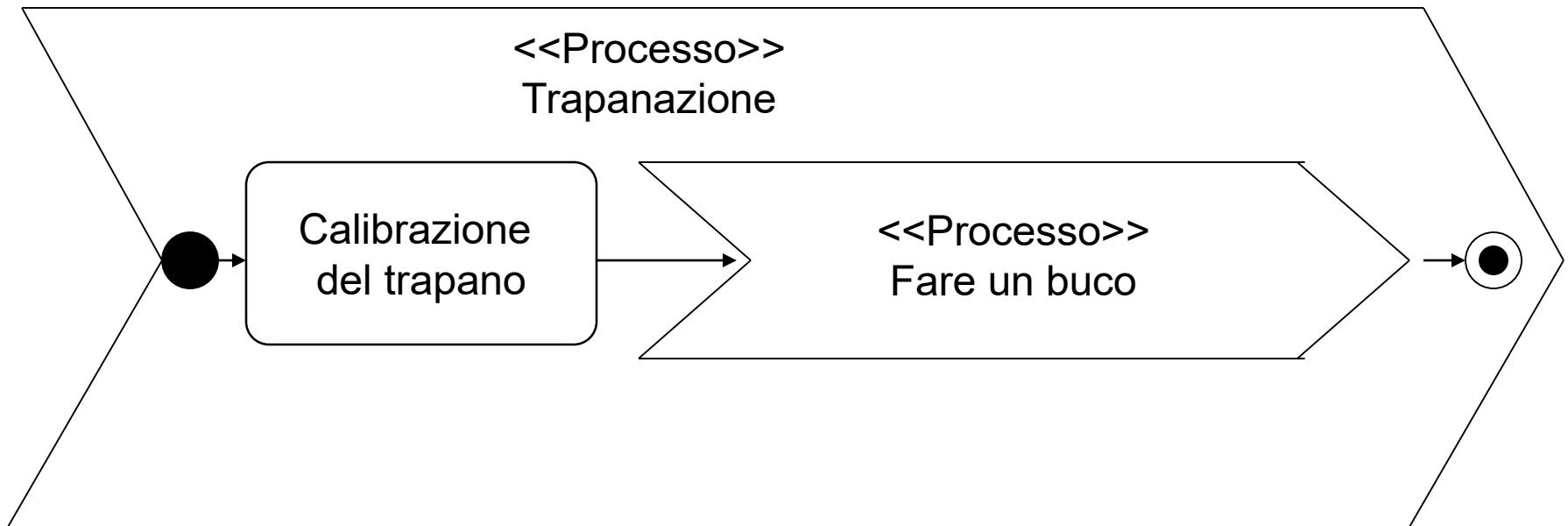
Activity Diagram: flusso di oggetti



Activity Diagram con processi



Activity Diagram con processi: esempio



Diagrammi UML

- **Activity Diagram**
- **Use Case Diagram**
- **Class Diagram**
- **Sequence Diagram**
- **Collaboration Diagram**
- **Statechart Diagram**



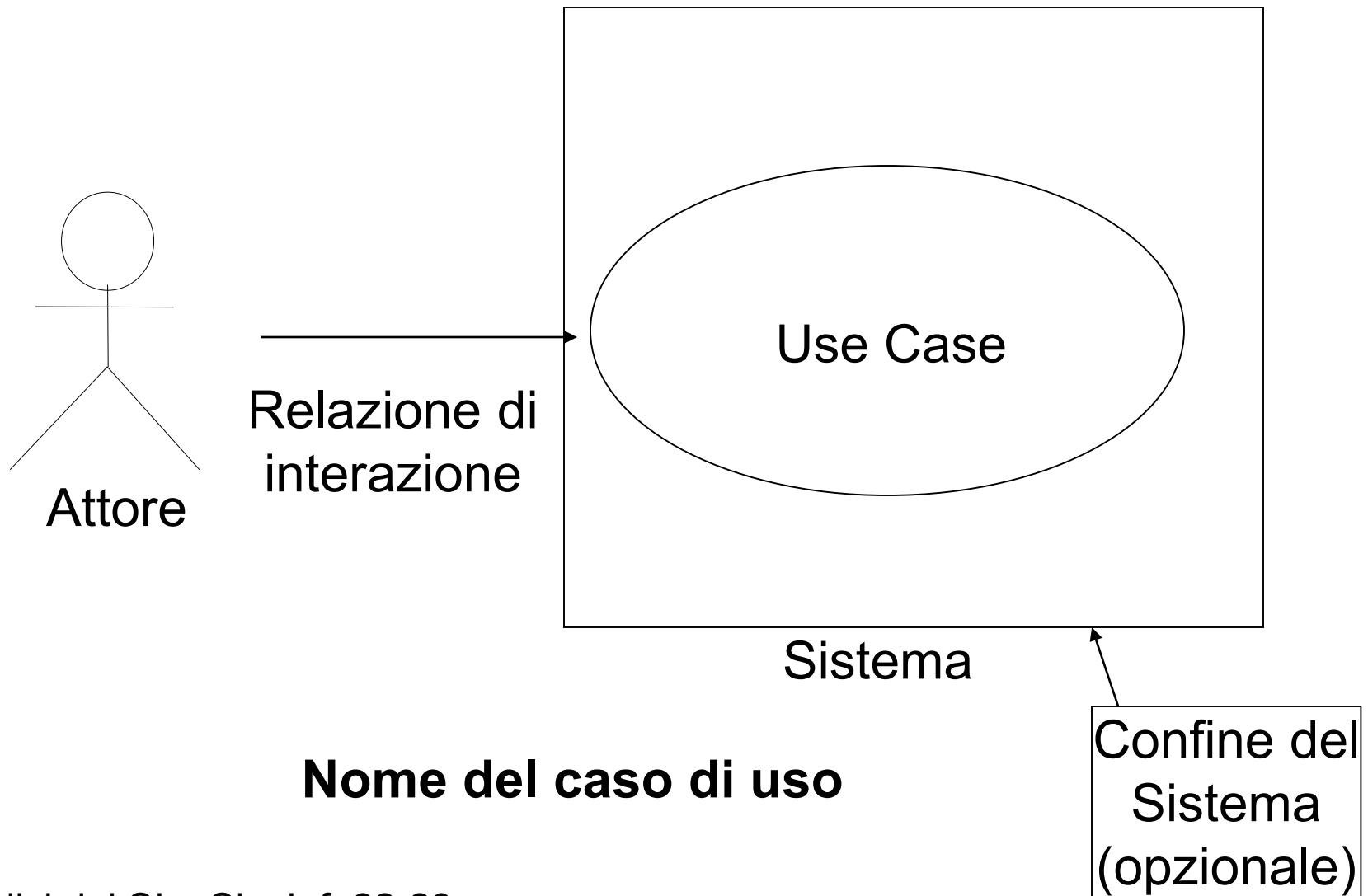
Lo Use Case (caso d'uso)

- Lo use case è un contratto
- Descrive l'interazione fra due entità che interagiscono fra loro
- Consente di stabilire :
 - **Servizi forniti**
 - **Servizi richiesti**
 - **Utenti abilitati**
 - **Vincoli nell'erogazione**

Lo Use Case (caso d'uso)

- Il primo scopo è quello di trovare un confine preciso (boundary) per il sistema/sottosistema/componente che si sta analizzando
- Una volta definito il confine si può stabilire
- **cosa fa il sistema rispetto all'esterno**
- **e identificare attori e use case**

Use Case Diagram: sintassi

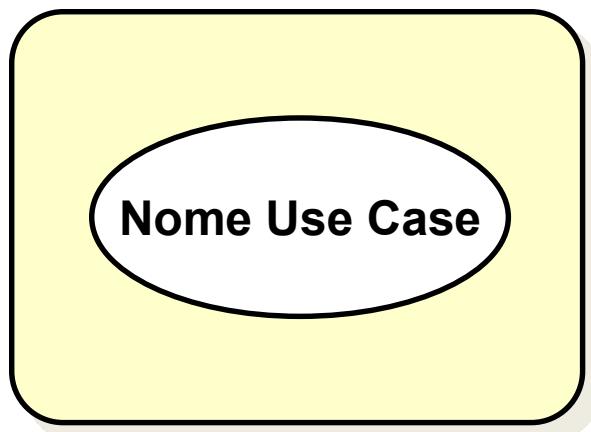


Definizione formale di uno Use Case

SEQUENZA DI TRANSAZIONI,
eseguita da un ATTORE
in interazione col SISTEMA,
la quale fornisce
un VALORE MISURABILE
per l'attore

**Definisce le richieste al sistema
dipendenti dall'attore**

Definizione pratica di Use Case



- Sequenza di transazioni in dialogo col sistema
- Comporta Sempre uno o più Attori
- Rappresenta COSA (non come) il sistema offre all'attore
- Mappato alle attività di business

**Gli Use case definiscono fabbisogni di sistema
“testabili” con una vista “fuori-dentro”**

Cos'è un attore

- Un attore rappresenta un'entità esterna al sistema (una persona, un altro sistema software, un componente hardware) che interagisce col sistema
- Un attore individua un ruolo piuttosto che un'entità fisica

Cos'è uno Use Case (caso d'uso)

- Uno use case rappresenta una situazione tipica di utilizzo del sistema e comprende in sé vari flussi possibili di esecuzione
- Uno use case rappresenta un'importante parte di funzionalità, completa dall'inizio alla fine

Come si trova un attore?

- Un attore è un **ruolo**
- Usare le domande:
 - chi ha bisogno del sistema? (il caro vecchio: cui prodest?)
 - chi userà le funzionalità principali?
 - chi dovrà manutenere e amministrare il sistema?
 - di quali dispositivi hardware il sistema ha bisogno?
 - con quali altri sistemi il sistema dovrà comunicare?

Come si trova uno use case?

Per ognuno degli attori precedentemente identificati, si può rispondere alle seguenti domande:

- quali funzioni l'attore richiede al sistema?
- l'attore ha bisogno di leggere o scrivere o immagazzinare informazioni nel sistema?
- l'attore deve ricevere notifiche di eventi dal sistema?

Attori e Use Case

- Attori e use case sono sempre collegati fra loro
- un attore isolato non può interagire col sistema
- uno use case isolato non fornisce alcuna funzionalità all'esterno (stiamo parlando di funzionalità che abbiano un senso all'esterno)

Attori e Use Case - 2

Un attore può essere

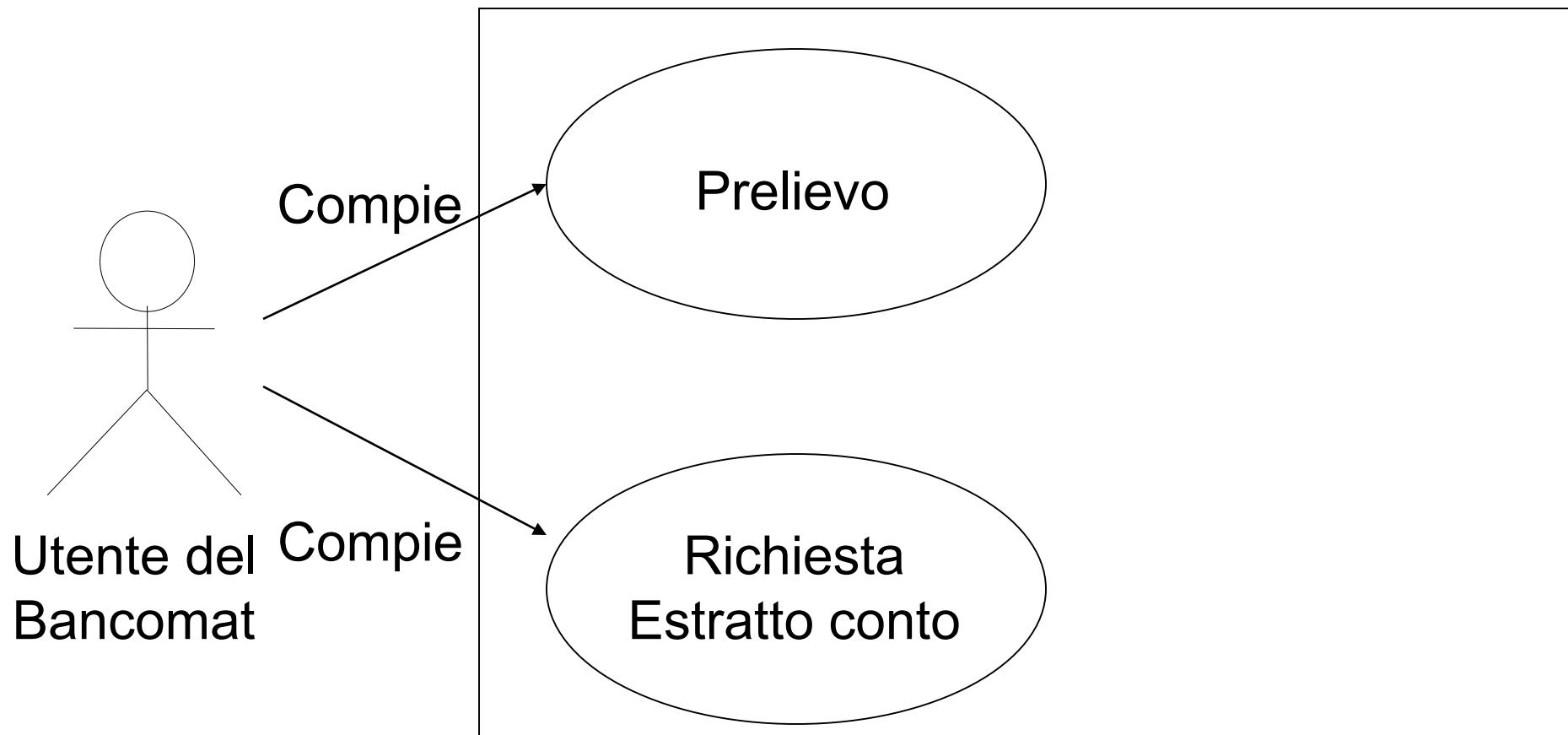
- **attivo**, ovvero inizia uno use case
- **passivo**, ovvero partecipa a uno use case, ma non lo inizia

Perché usare gli Use Case?

PER EVITARE :

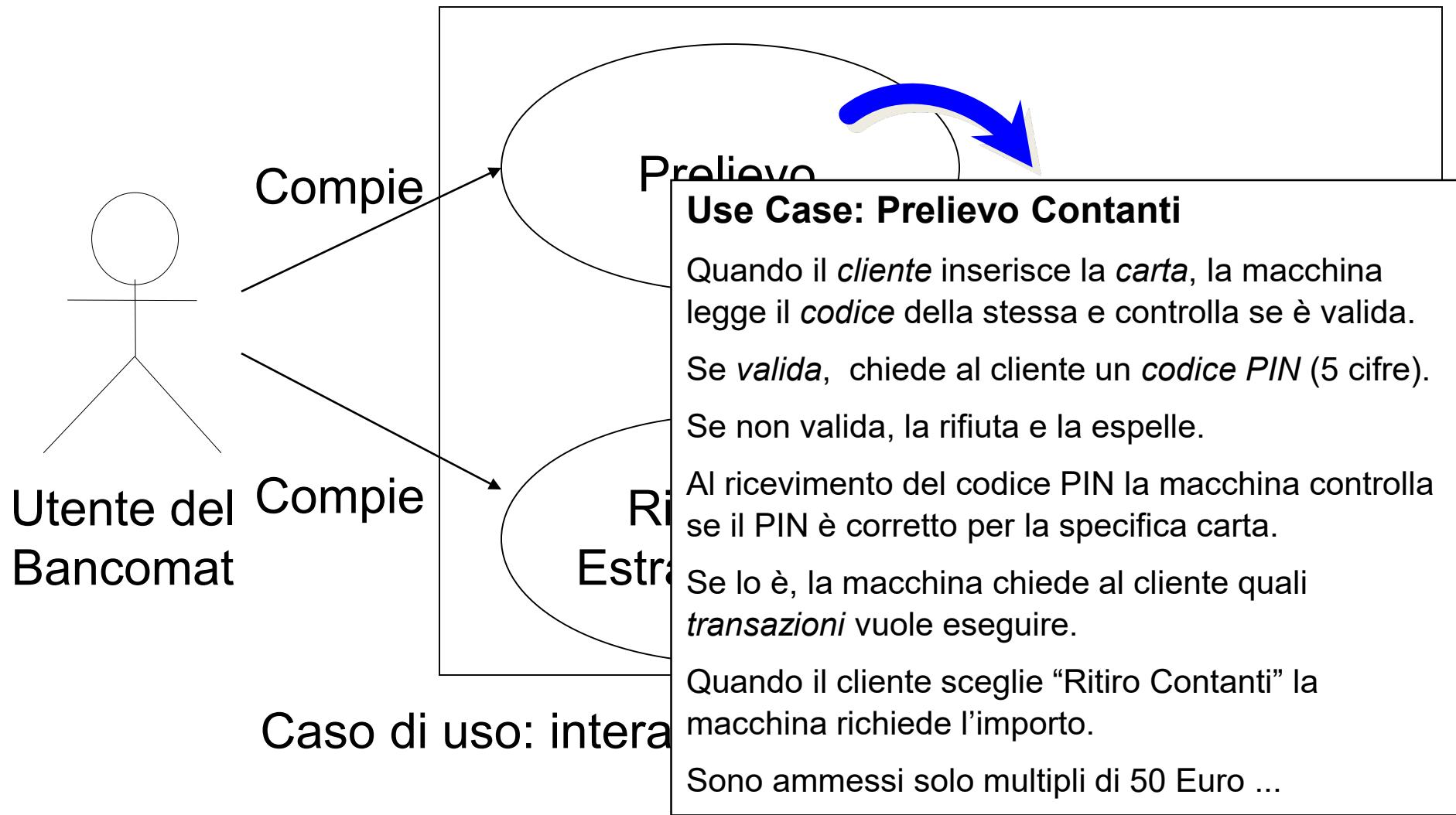
- *Utente*: non è ciò che volevo !
- *Analista*: ma come, fa tutto quello che mi hai chiesto...
- *Utente*: si, ma non come intendeva !

Use Case Diagram: esempio

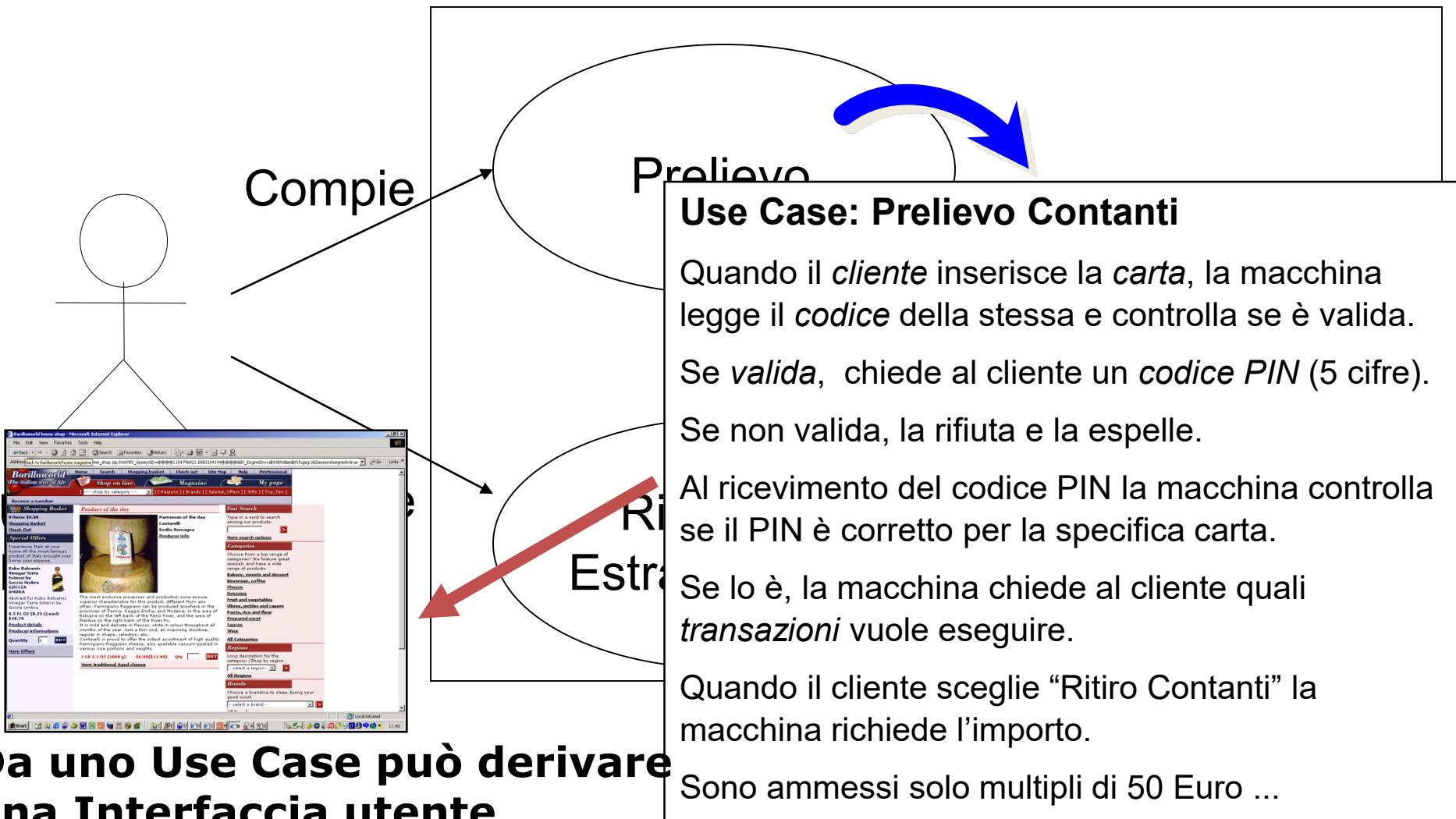


Caso di uso: interazioni con il bancomat

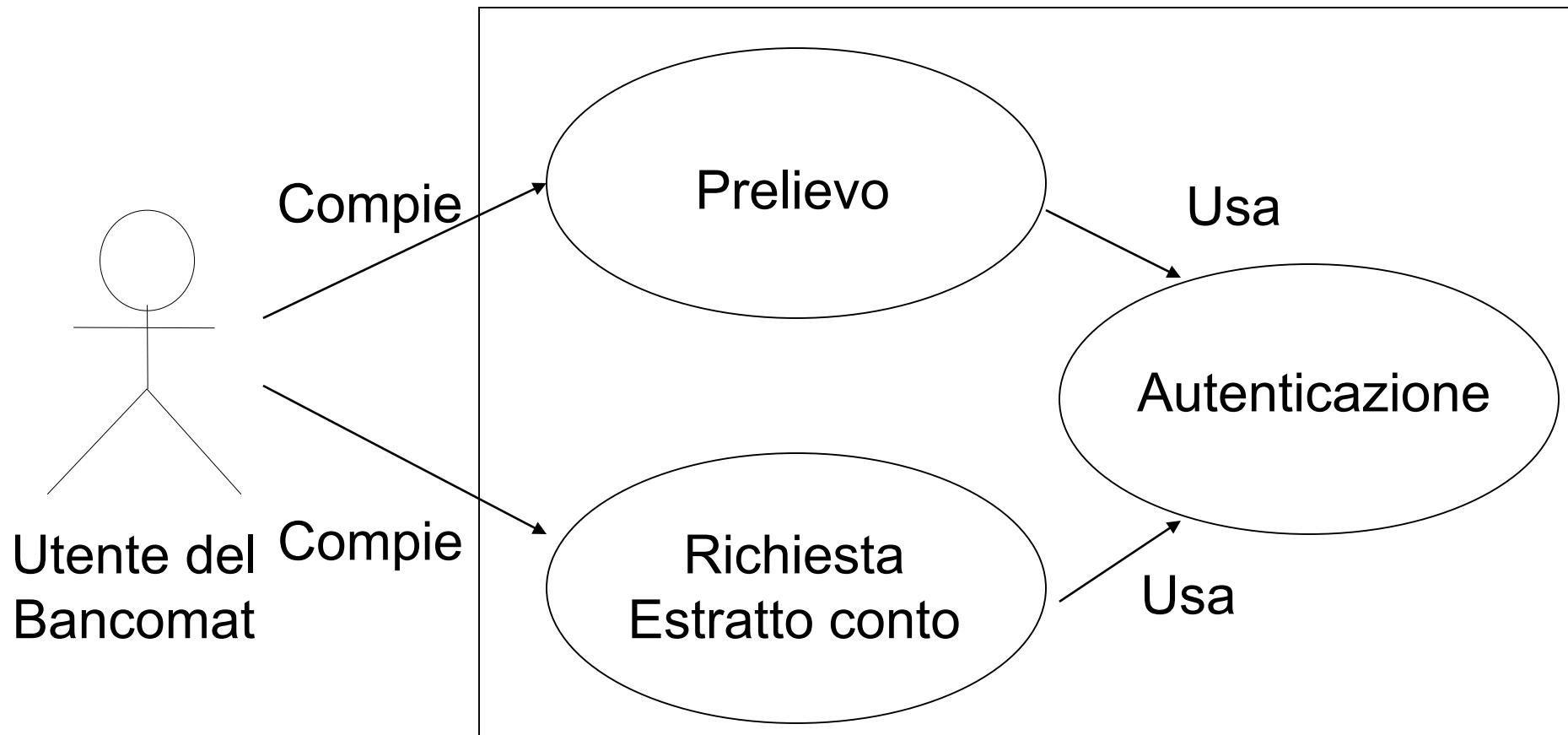
Descrizioni Use Case: specificano i dettagli



Use Case Diagram e Interfacce Utente

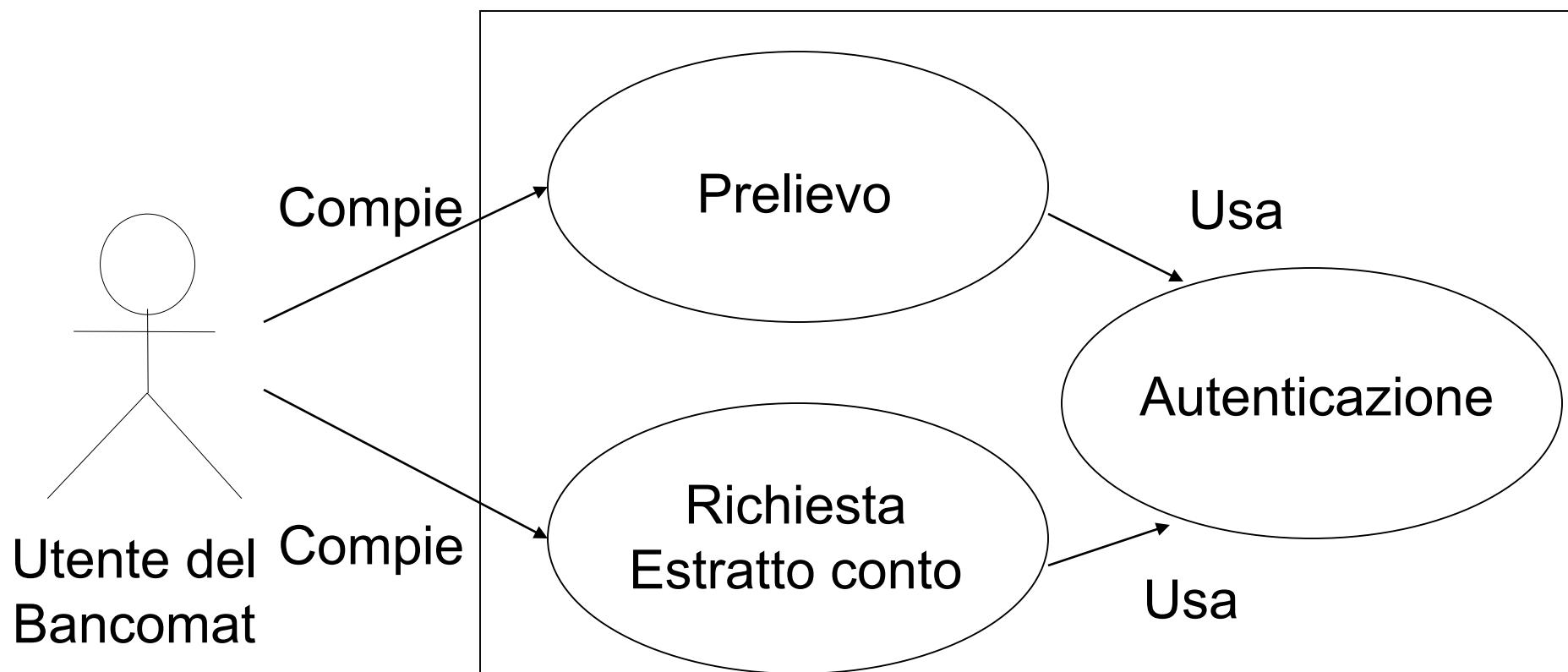


Use Case Diagram: scomposizione



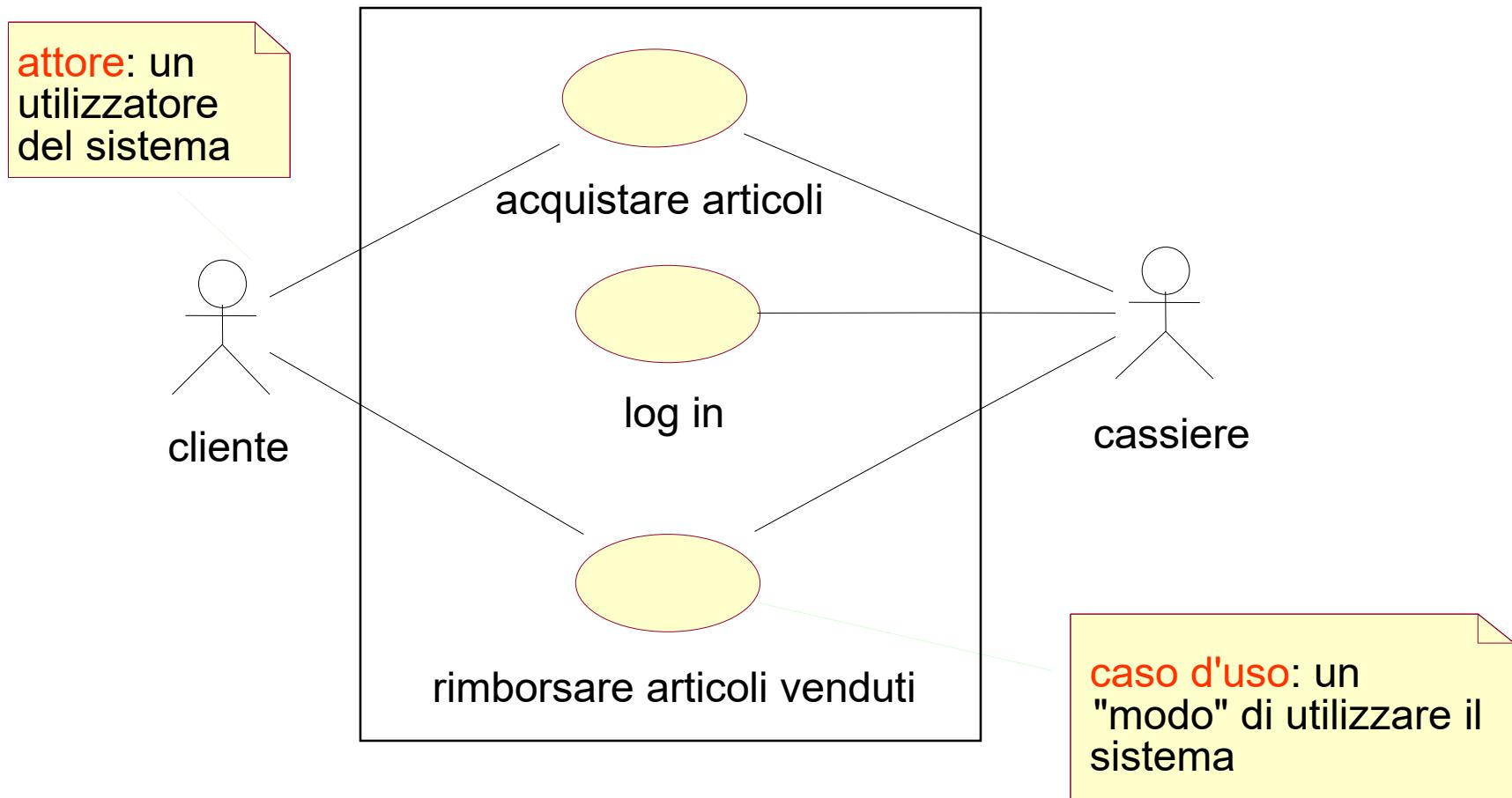
Caso di uso: interazioni con il bancomat

Modello dei rapporti tra Attori e Use Case

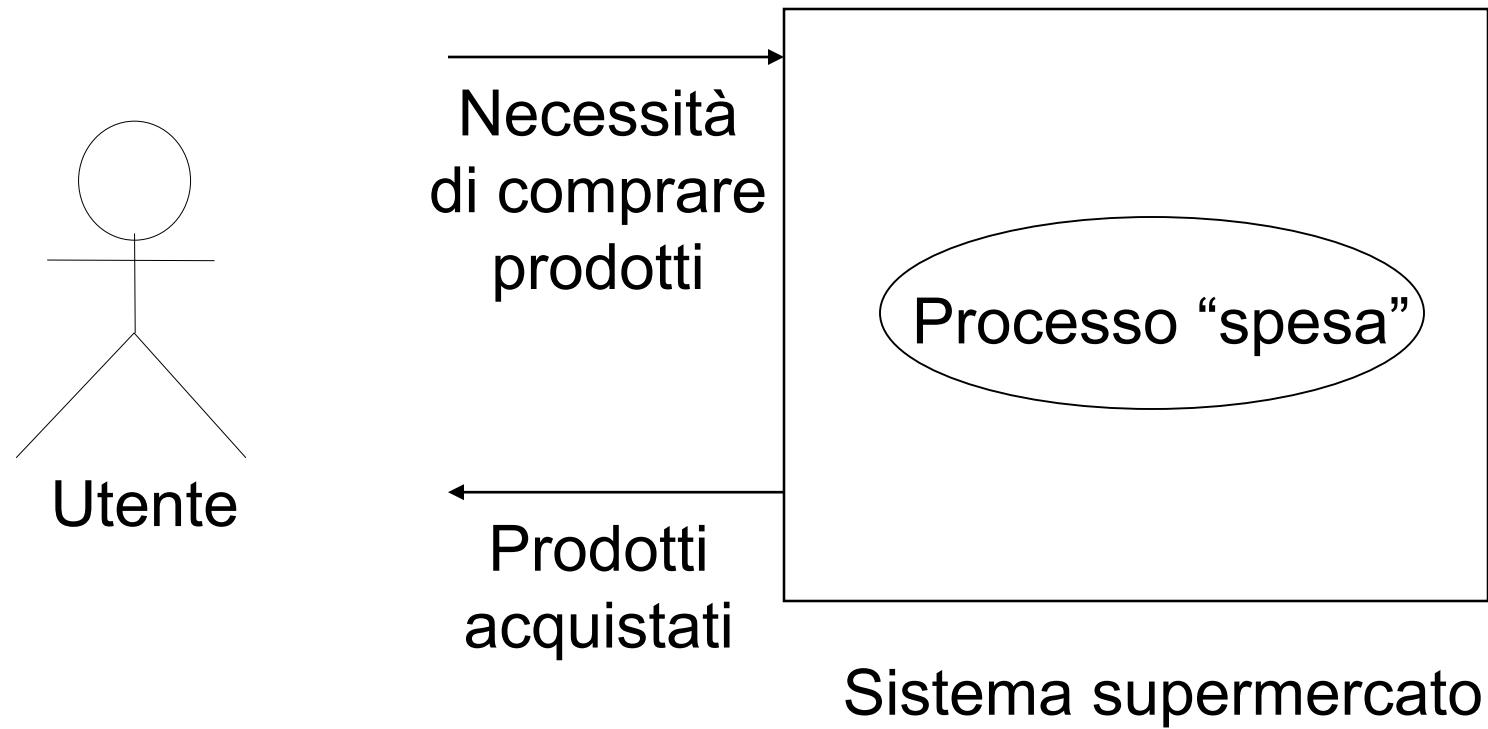


- ◆ **Attori rappresentati da una persona**
 - ◆ **Use case = ellissi**
 - ◆ **Astrazione delle interazioni tra Attori ed uno use case**
- 4 - Analisi dei **Il sistema è circondato dal riquadro**

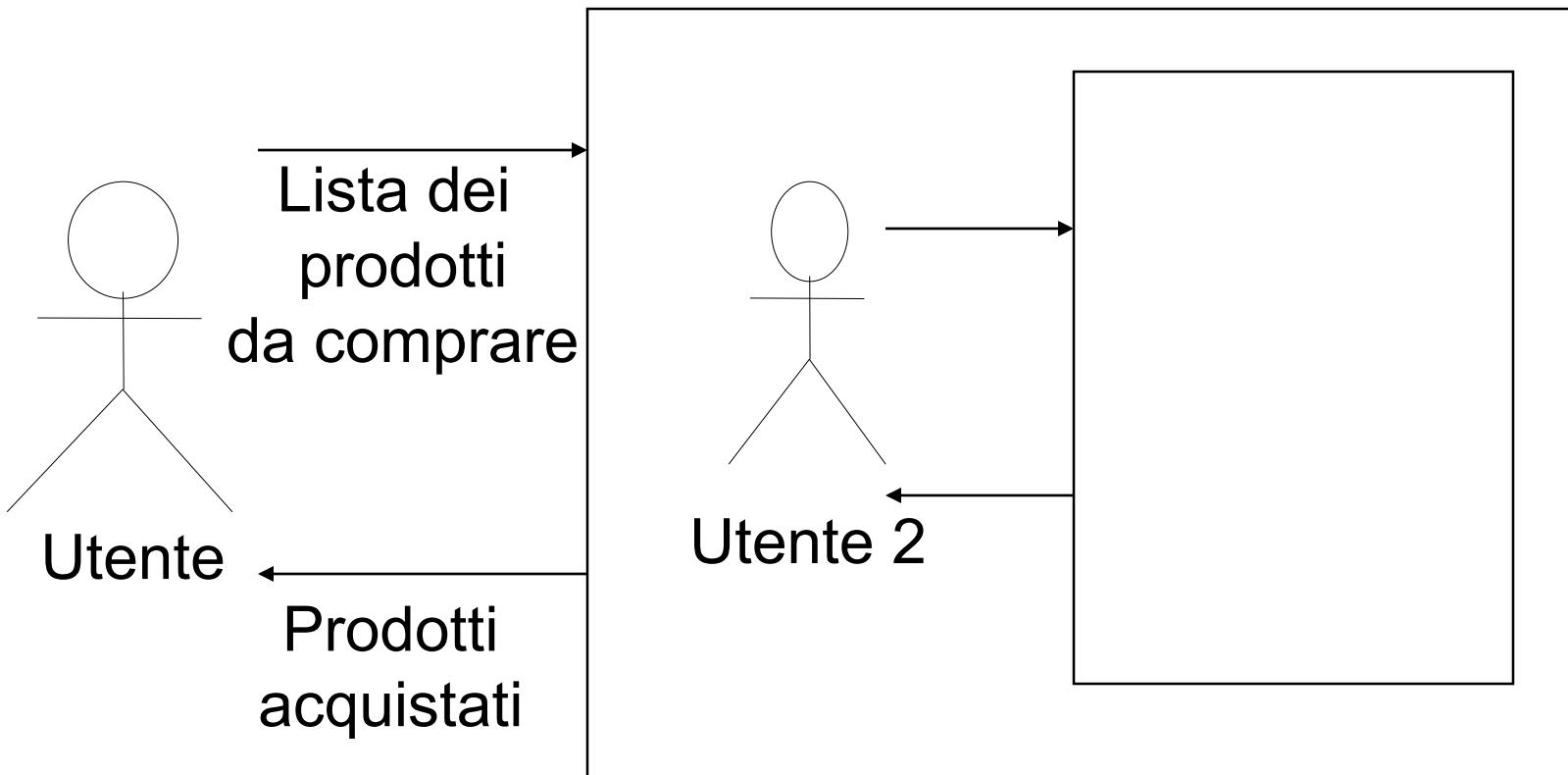
Esempio di Use Case: la cassa



Un processo del mondo reale: la spesa

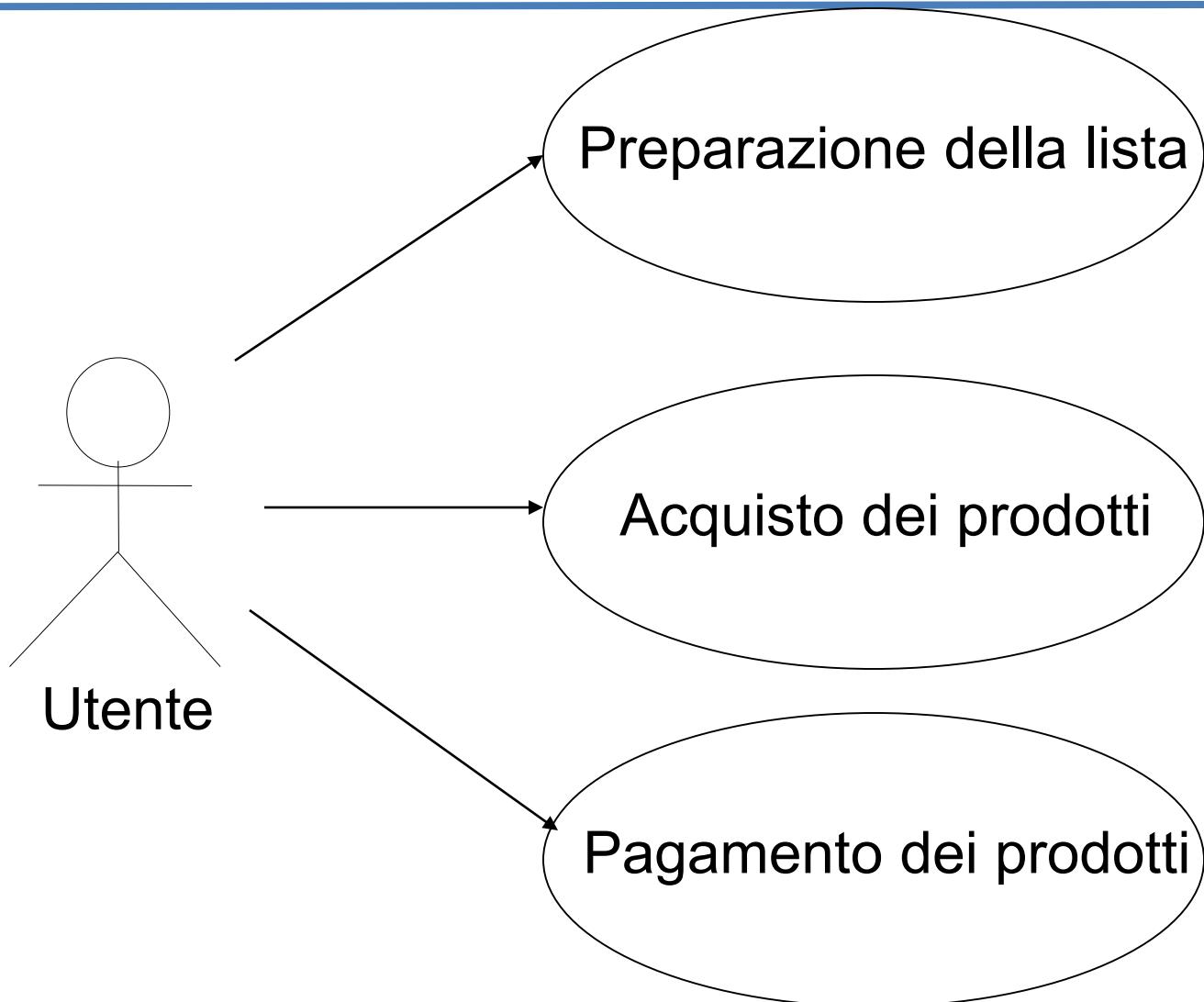


Quali sono i limiti di un sistema?

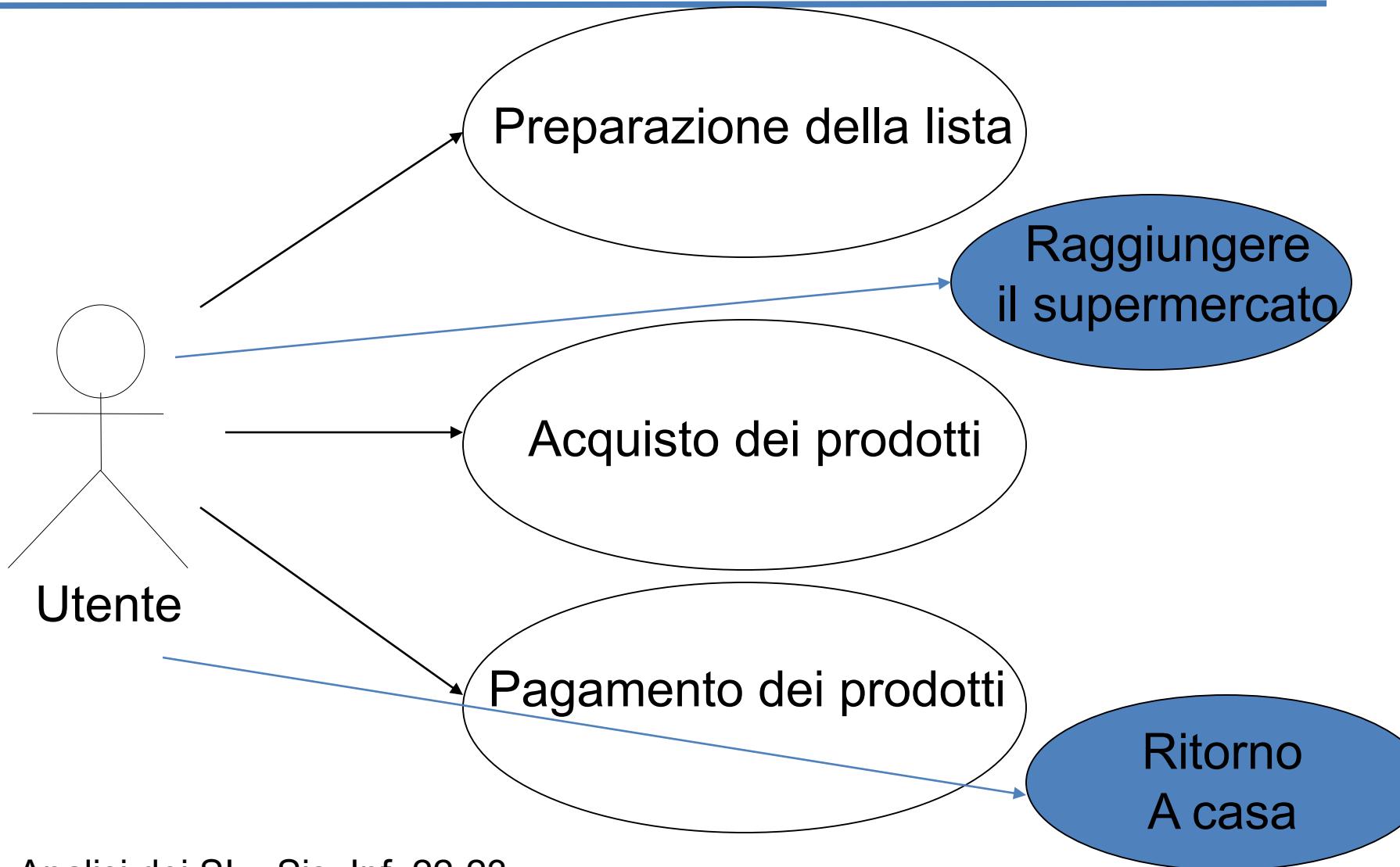


Sistema: utente 2 + negozio
Processo: “fai la spesa”

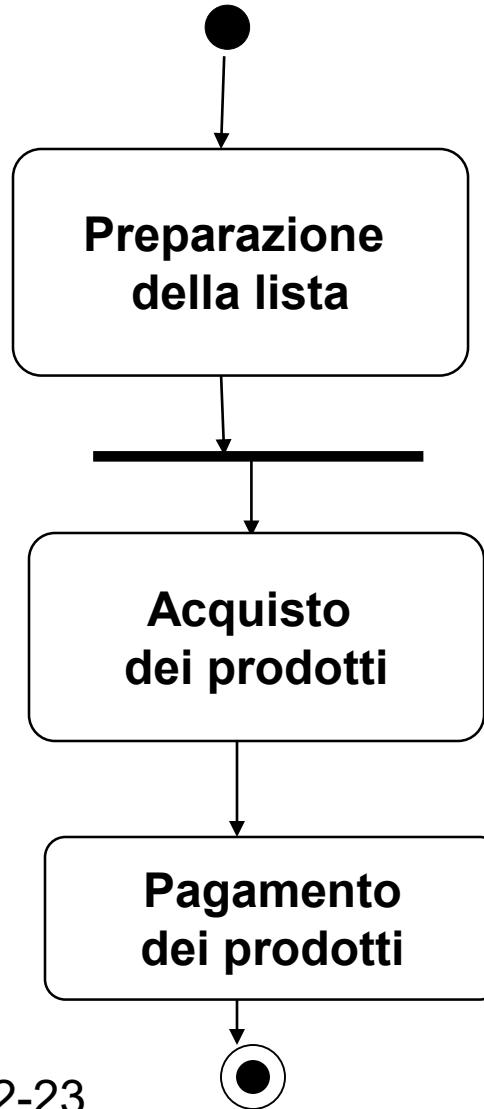
Le funzionalità della spesa: use case



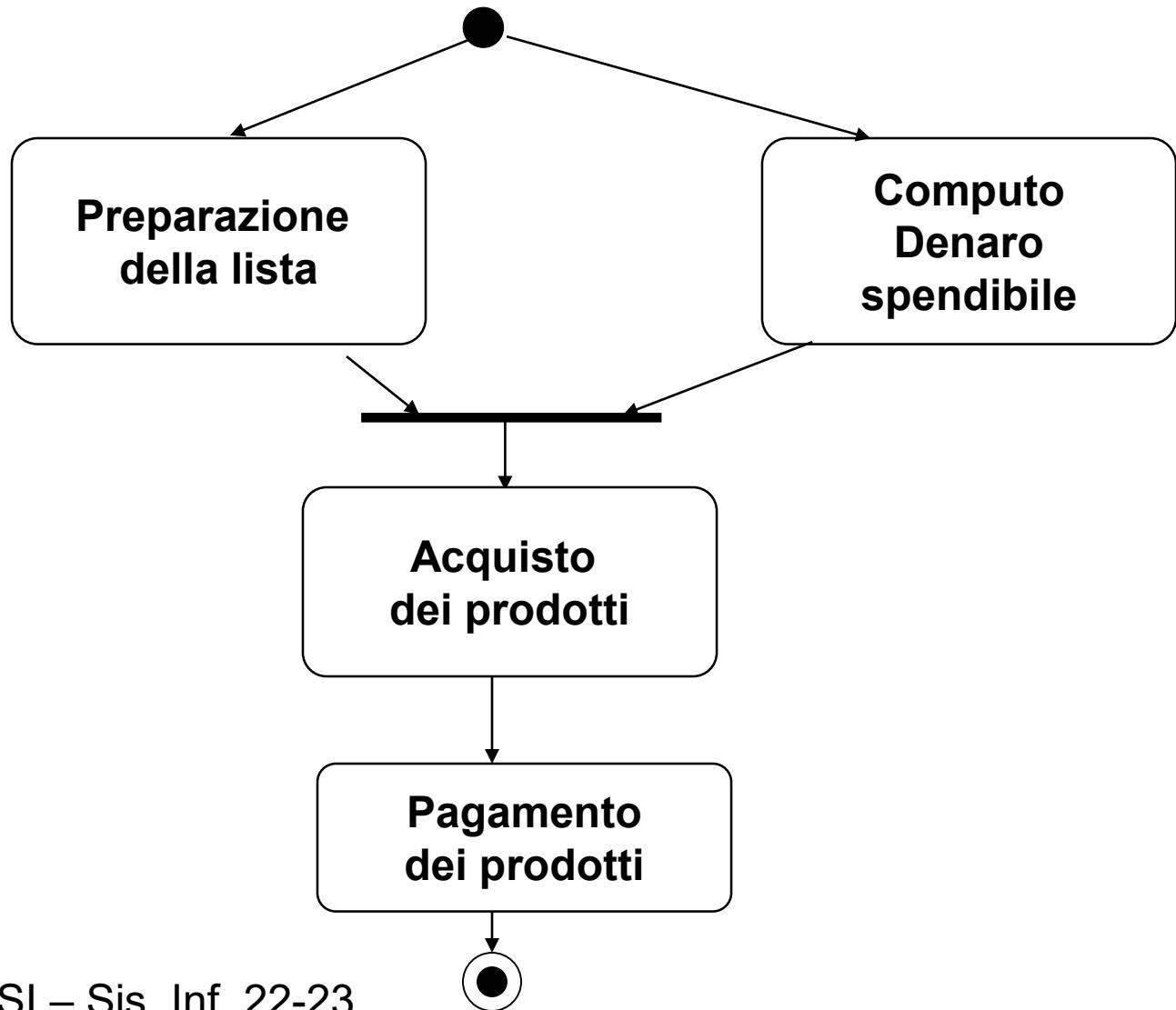
Le funzionalità della spesa: use case



Spesa: Diagramma di attività 1



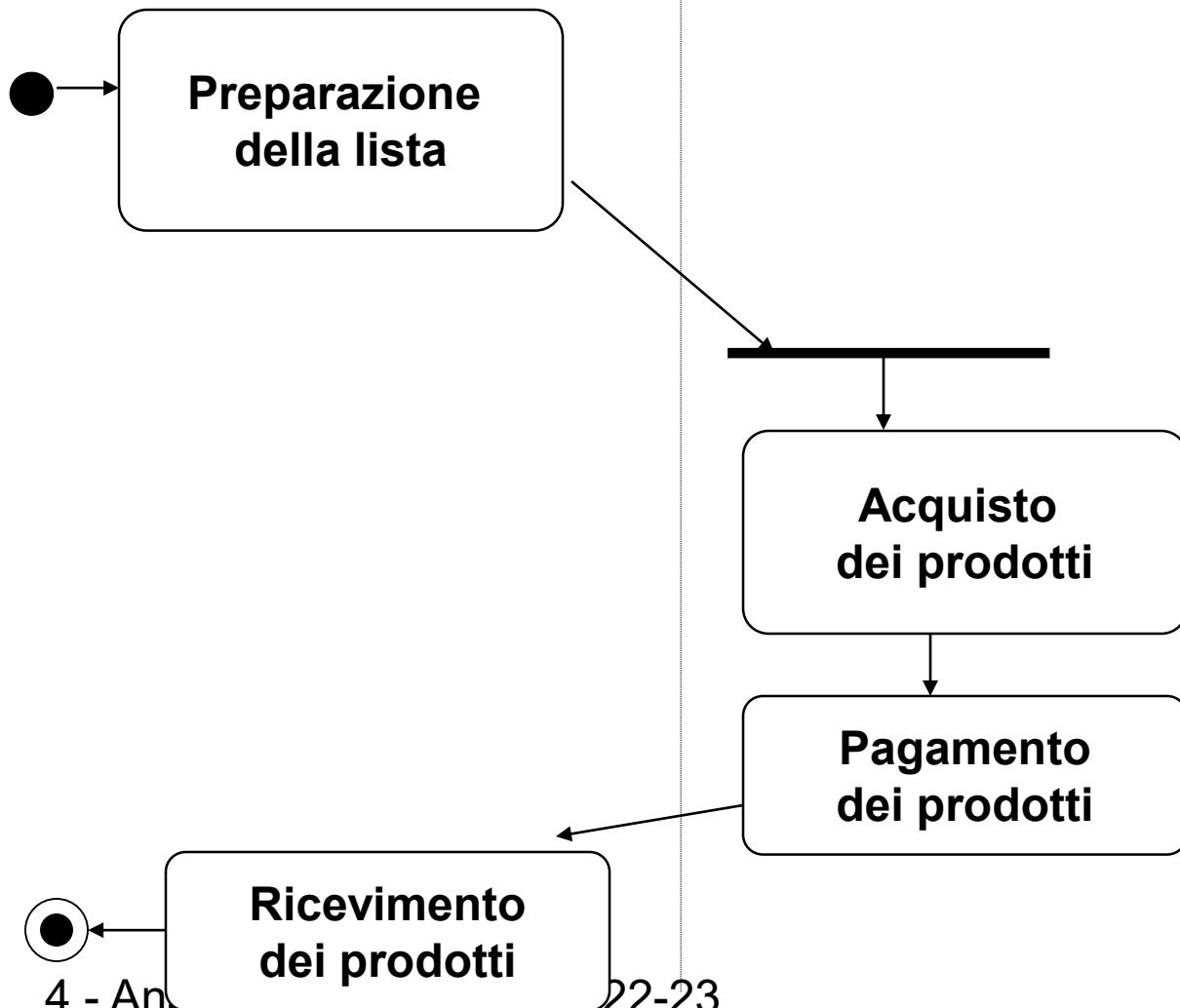
Spesa: Diagramma di attività 1



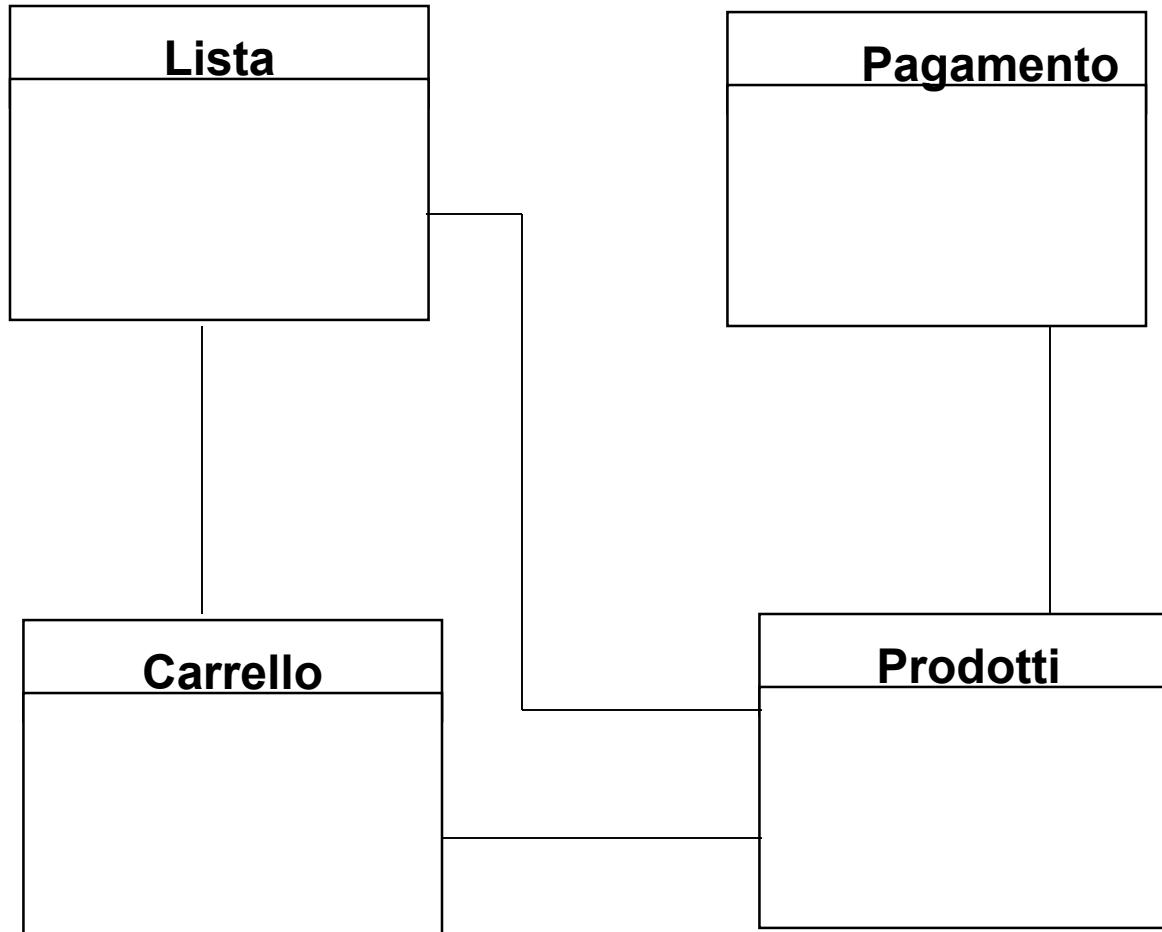
Spesa: Diagramma di attività 2

Utente 1: capo

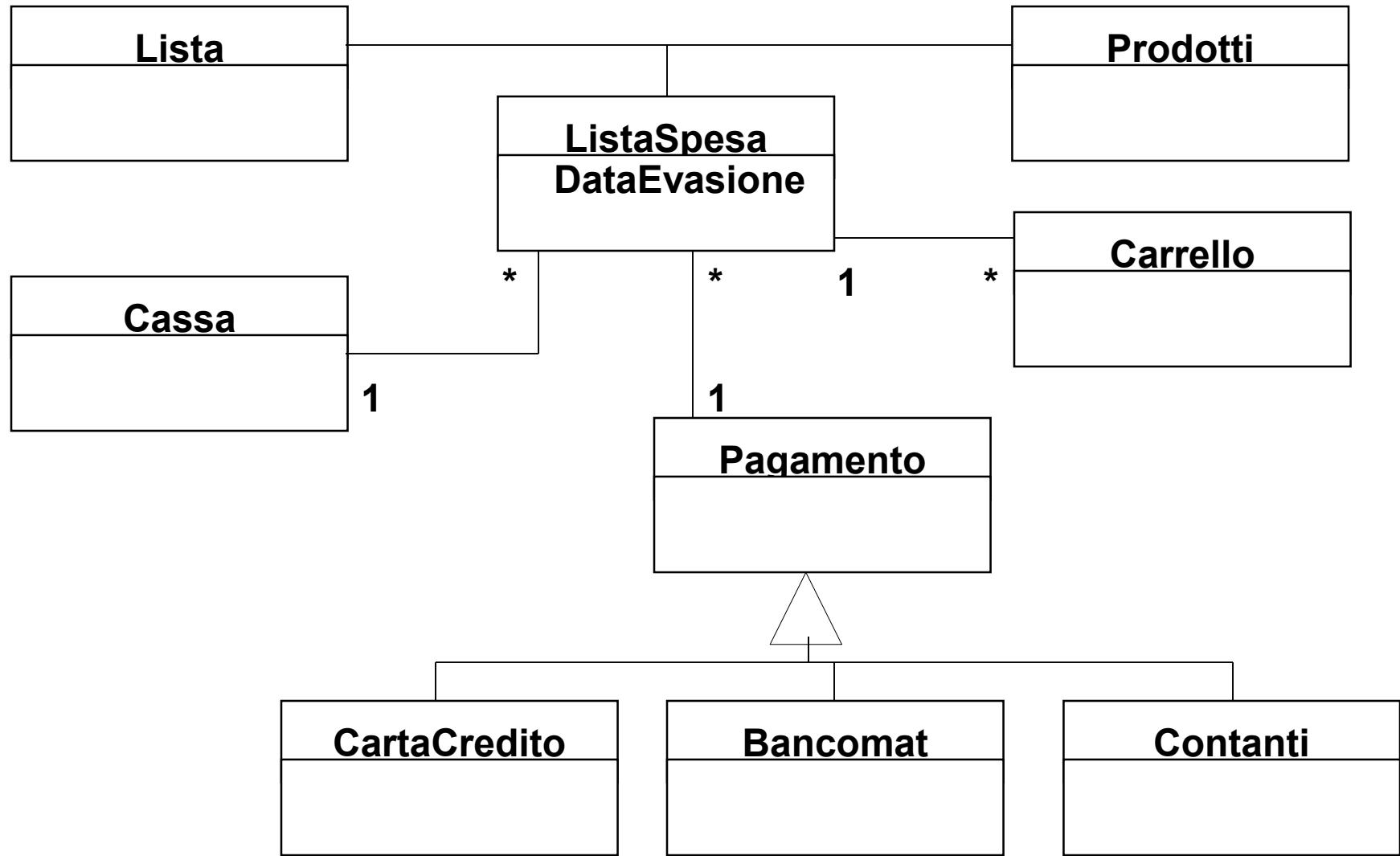
Utente 2: esecutore



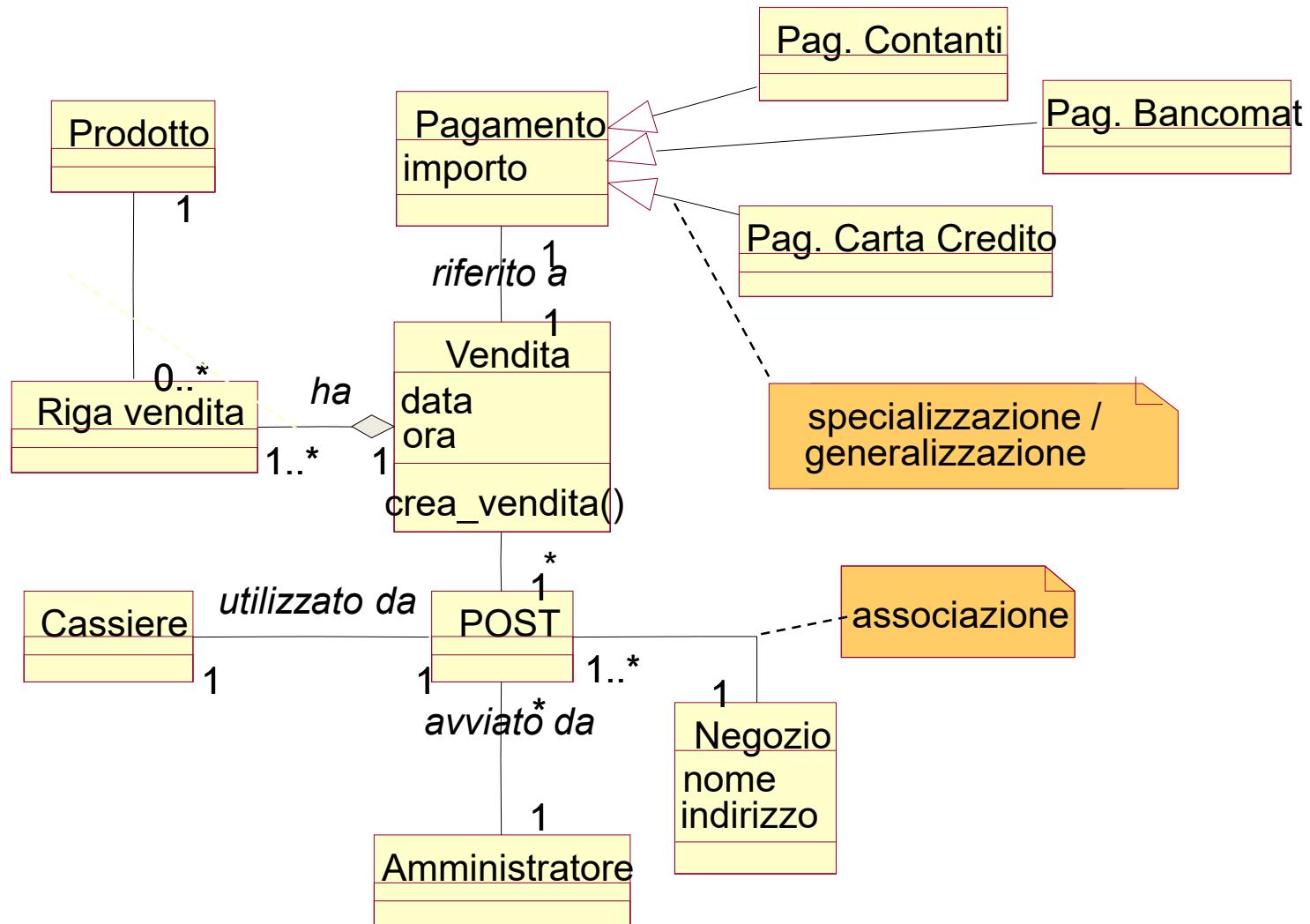
Spesa: cercare le classi



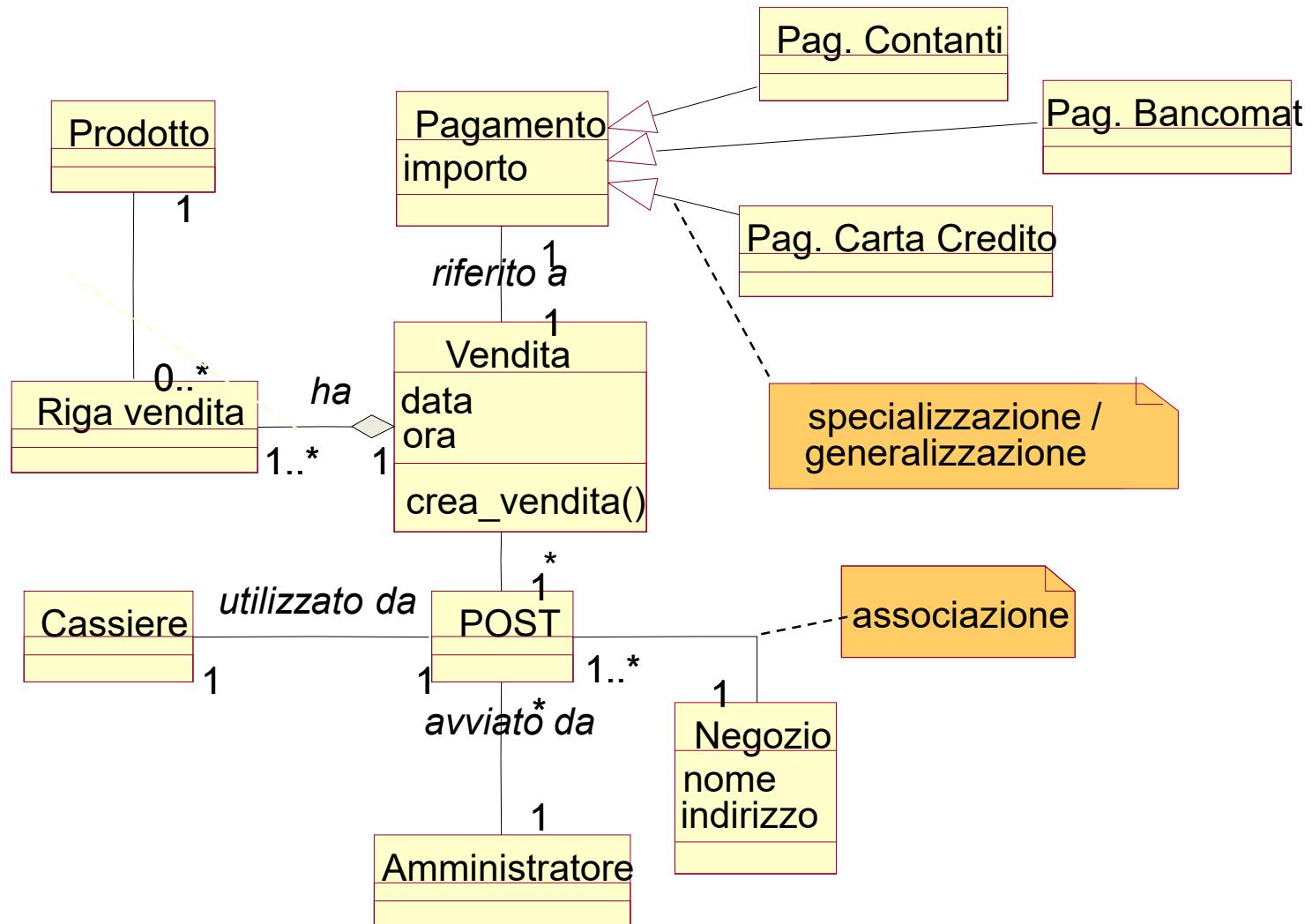
Spesa: Diagramma delle classi



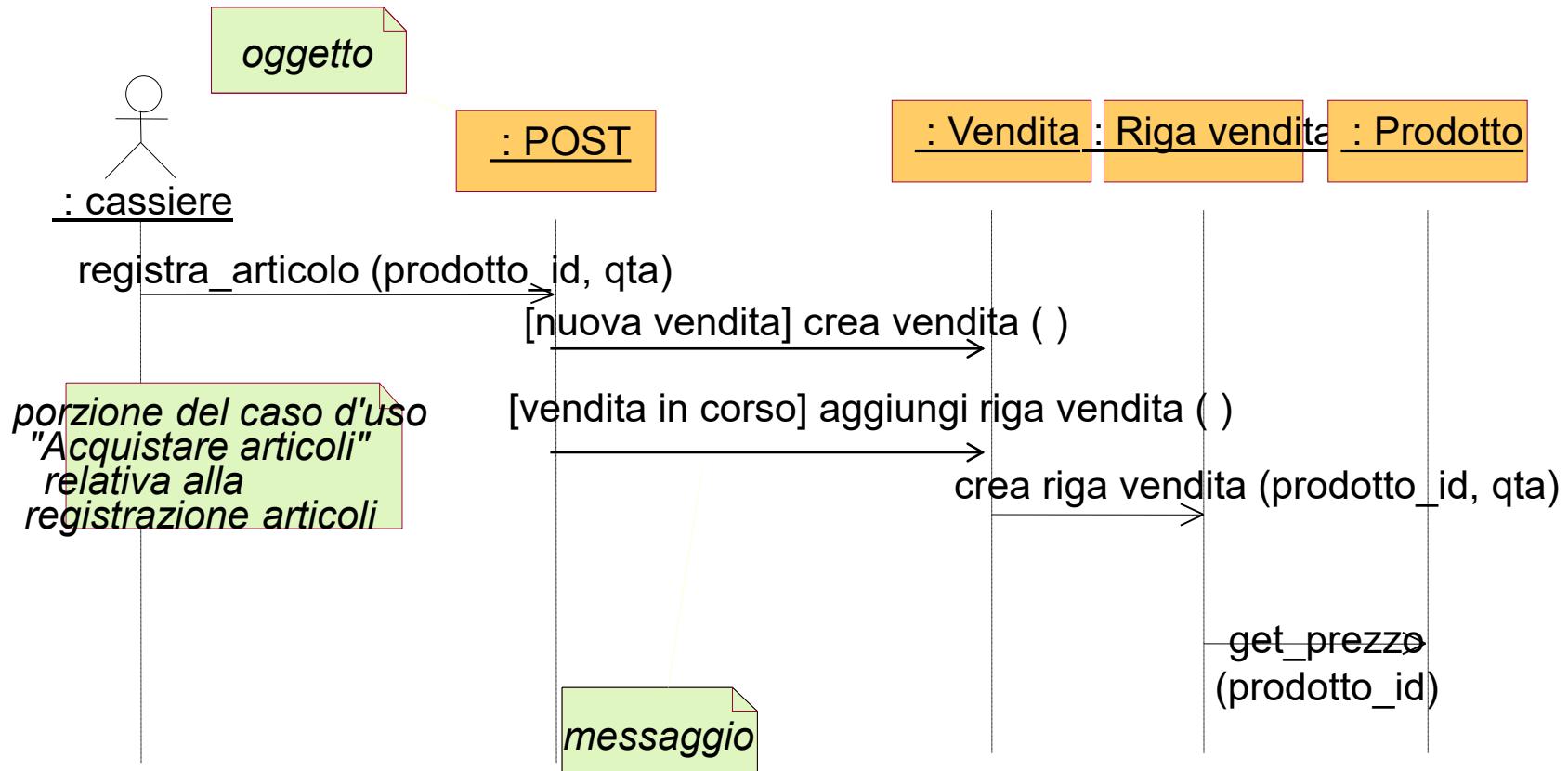
Spesa: Class Diagram completo



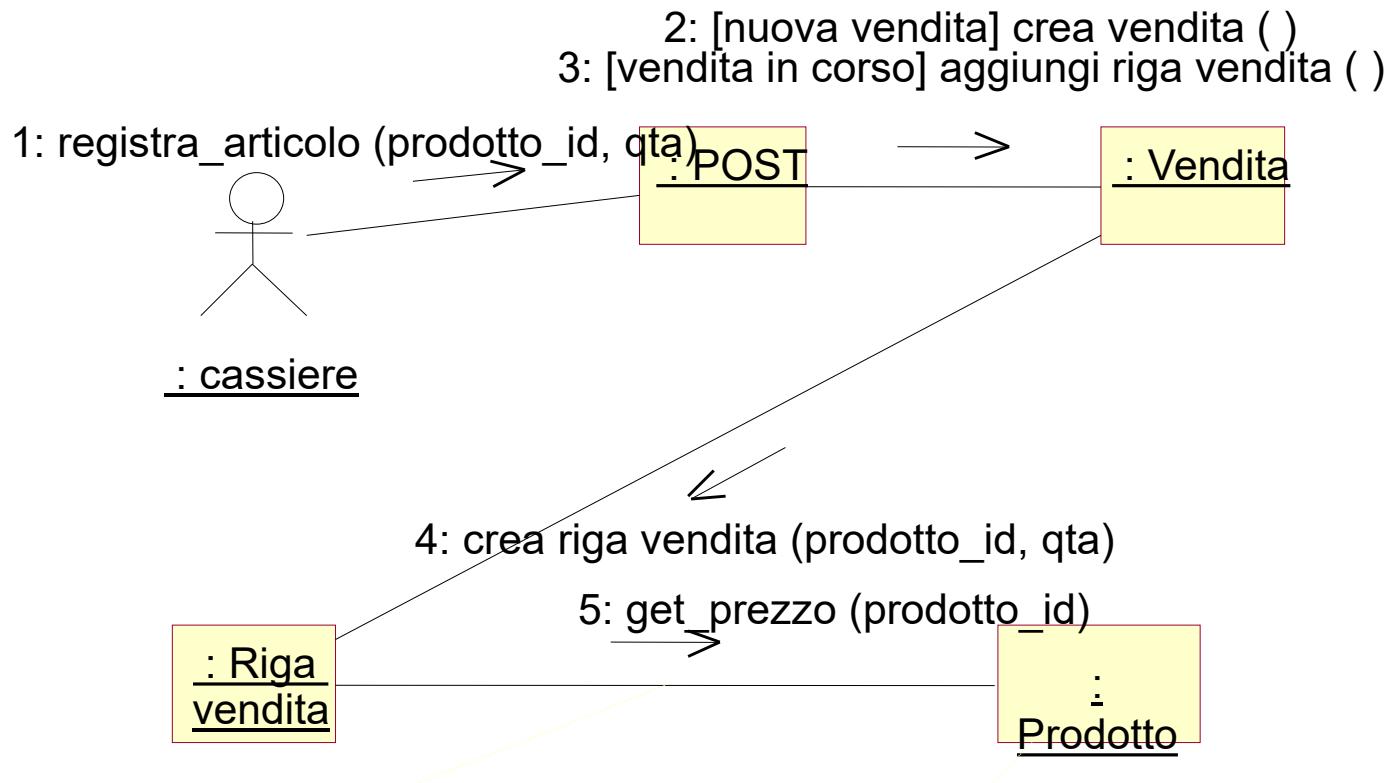
Spesa: Class Diagram completo



Spesa: diagramma di sequenza



Spesa: collaborazione

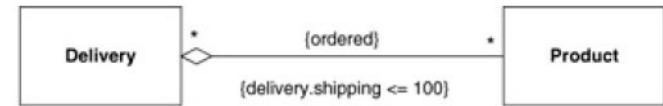


Eriksson-Penker Business (UML) Extensions

Formano un framework di base di estensioni business a UML, utilizzandone i meccanismi standard di estensione:

- **Stereotipi**: un'estensione al vocabolario di UML che consente di creare nuovi blocchi da quelli esistenti ma specifici per il proprio problema
- **Valori "taggati" (proprietà)**: estensioni delle proprietà di un elemento UML che consente di creare nuove informazioni nella specifica dell'elemento
- **Vincoli**: un'estensione della semantica di un elemento UML che permette di aggiungere nuove regole o modificare quelle esistenti

<<Physical>>
Engine

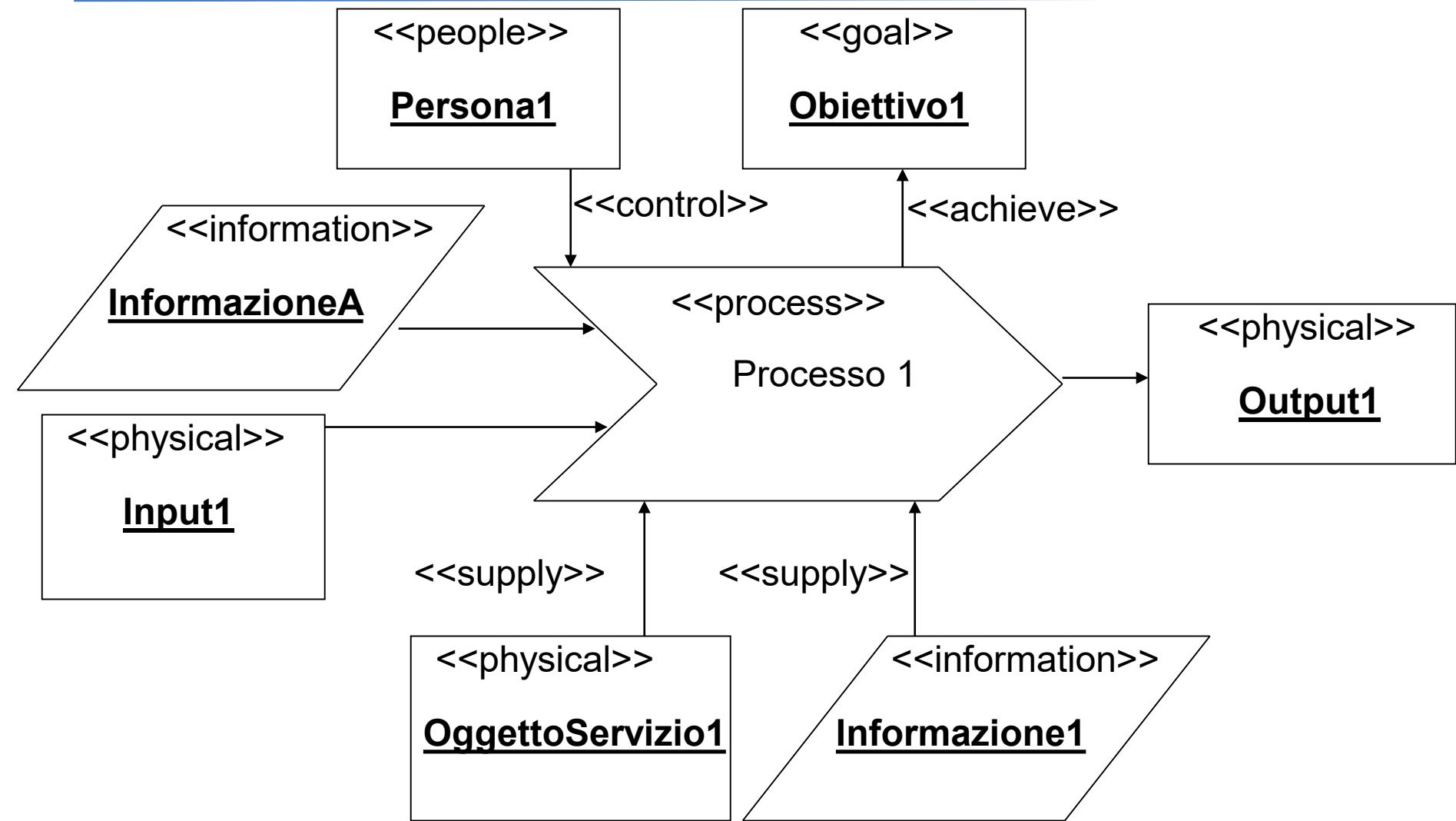


Eriksson-Penker Business (UML) Extensions

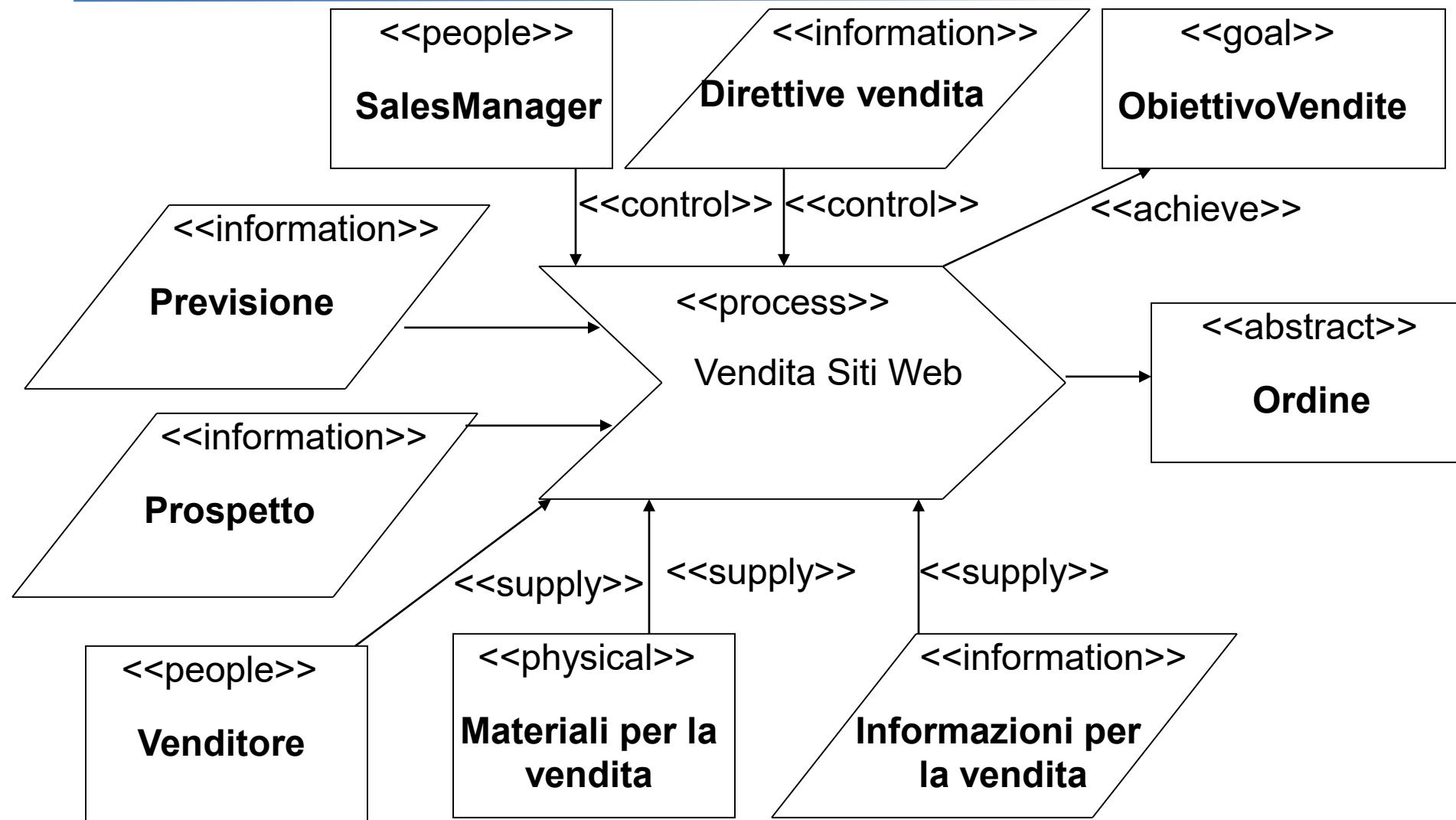
Concetti principali:

- **Risorse**: gli oggetti all'interno del business, come persone, materiali, informazioni e prodotti che sono utilizzati o prodotti nell'azienda. Le risorse sono manipolate (usate, consumate, raffinate o prodotte) attraverso i processi. Le risorse possono essere categorizzate in fisiche, astratte e informative (ognuna con il proprio stereotipo)
- **Processi**: le attività eseguite all'interno del business in cui lo stato delle risorse del business cambia. I processi descrivono come viene fatto il lavoro all'interno del business. I processi sono governati da regole
- **Obiettivi**: lo scopo del business, o il risultato che l'azienda nel suo complesso sta cercando di raggiungere. Gli obiettivi possono essere suddivisi in sotto-obiettivi e assegnati a parti individuali dell'azienda, come processi o oggetti. Gli obiettivi esprimono gli stati desiderati delle risorse e sono raggiunti dai processi. Gli obiettivi possono essere espressi come una o più regole
- **Regole**: una dichiarazione che definisce o vincola alcuni aspetti del business e rappresenta la conoscenza del business. Regola come il business dovrebbe essere eseguito (cioè, come i processi dovrebbero essere eseguiti) o come le risorse possono essere strutturate e collegate tra loro. Le regole possono essere imposte all'azienda dall'esterno da regolamenti o leggi, o possono essere definite all'interno dell'azienda per raggiungere gli obiettivi

Activity Diagram: il processo generico



Esempio di AD: vendita di siti Web



Un esempio di goal: l'obiettivo vendite espanso

<<goal>>

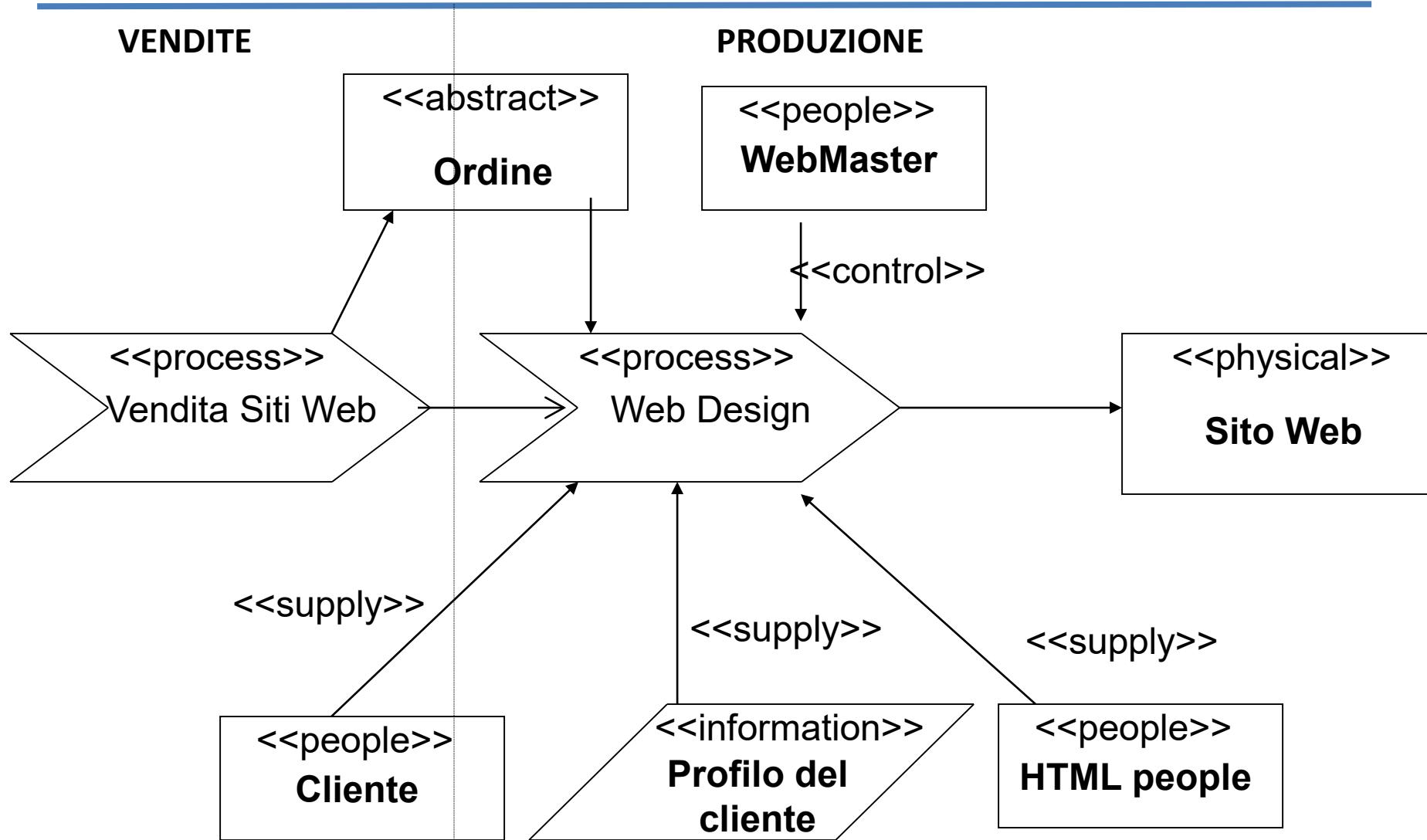
**ObiettivoVendite:
quantitativo**

fatturato: valuta = € 125.000,00

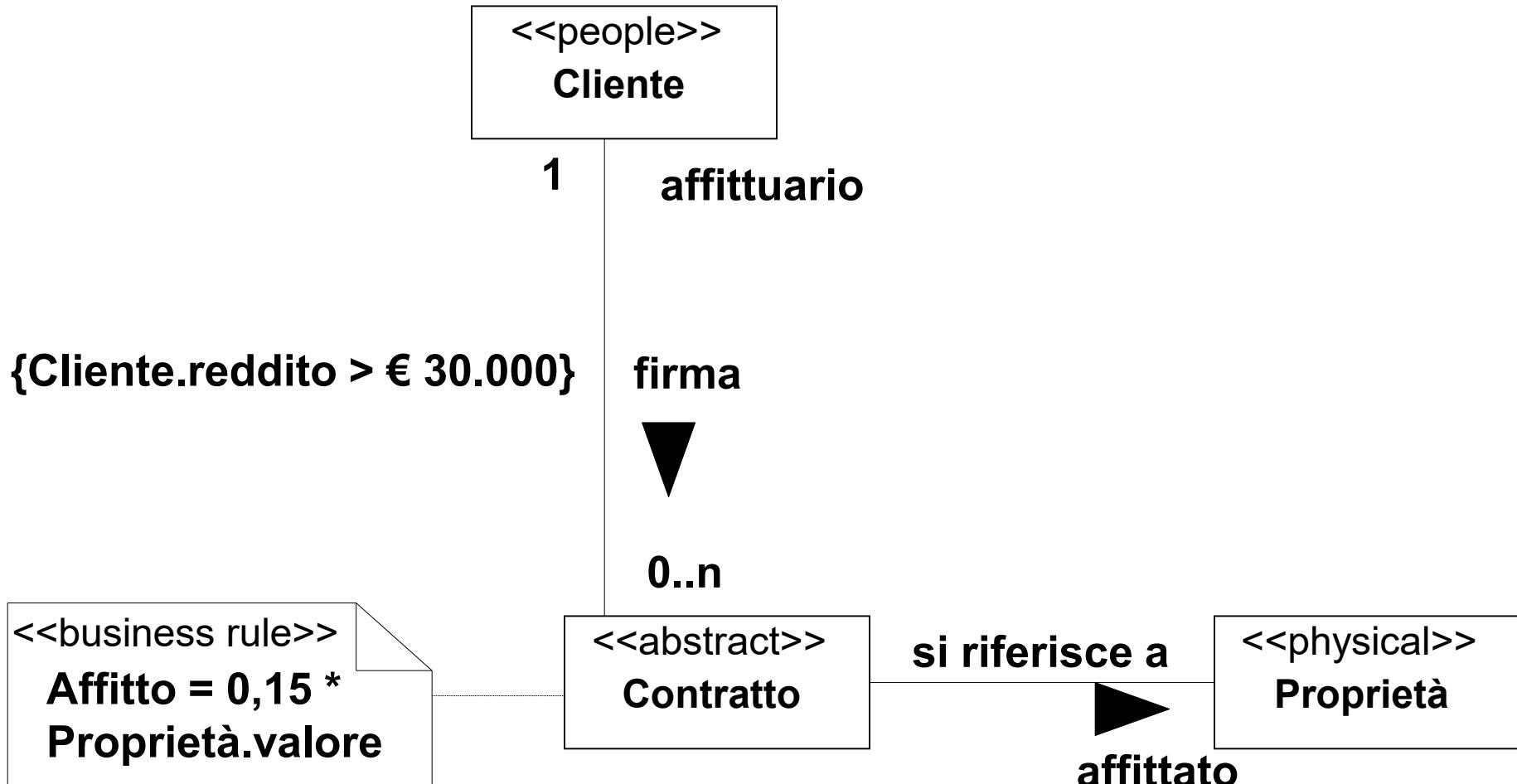
costi: valuta = € 75.000,00

scadenza: data = 31/12/2022

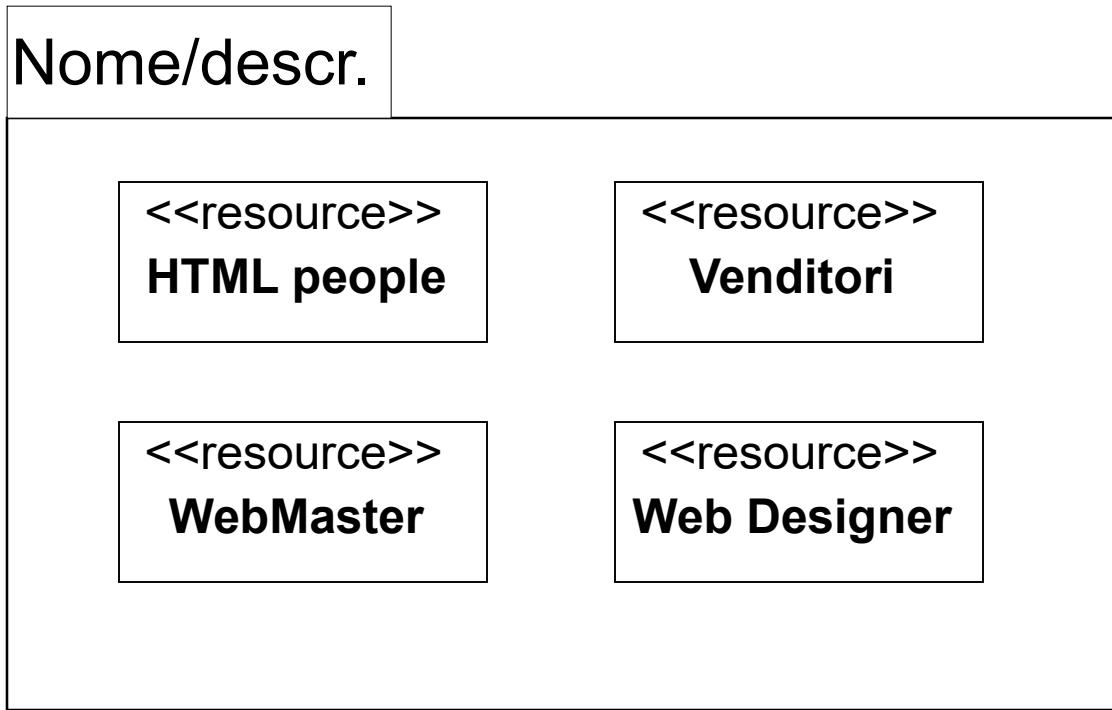
Combinazione AD/swimlanes



Class Diagram per regole business

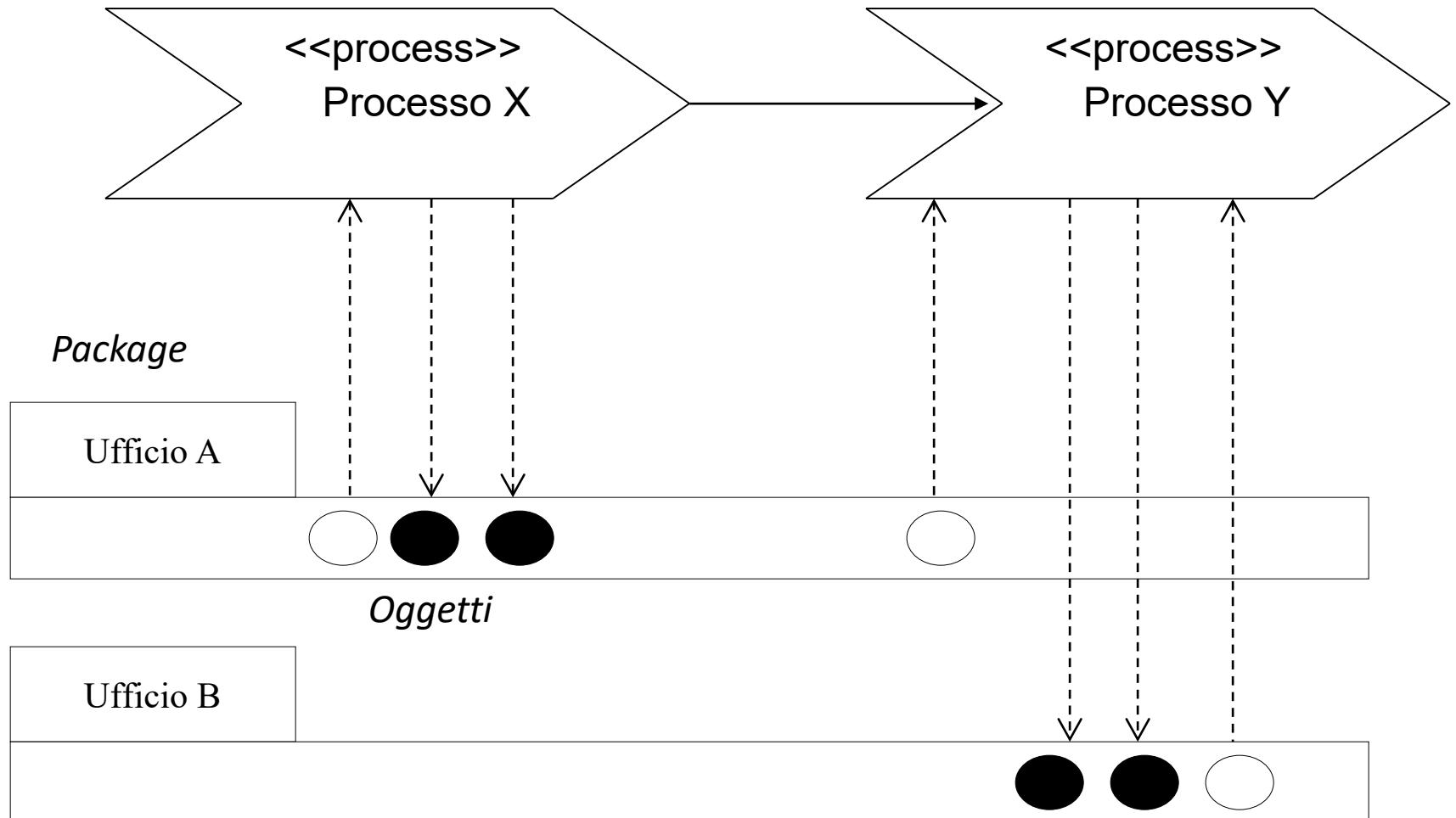


I Package di UML



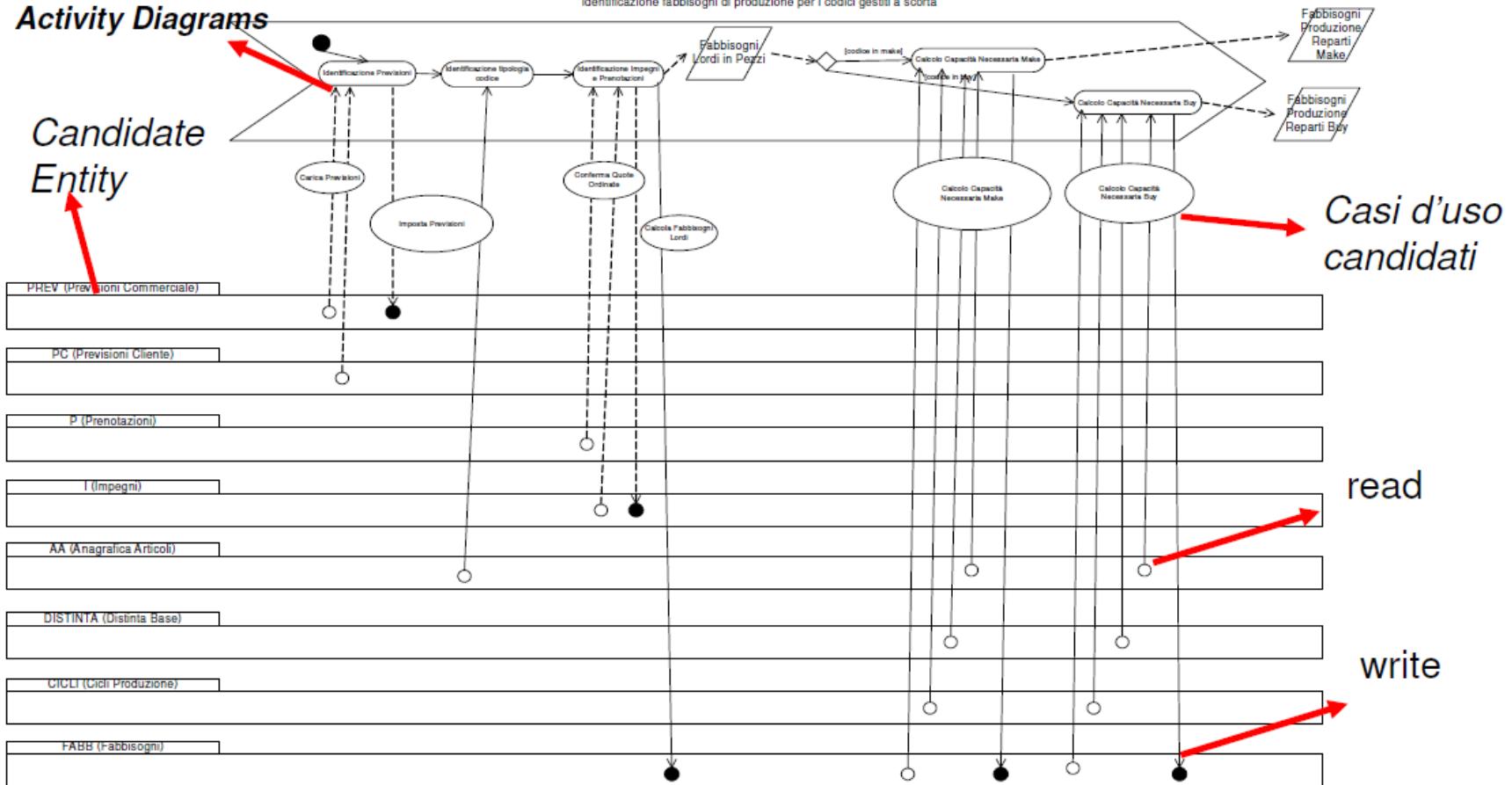
Un package è un raggruppamento generale di elementi correlati fra loro da un legame logico che il modellatore ritiene importante

Assembly line



Assembly lines

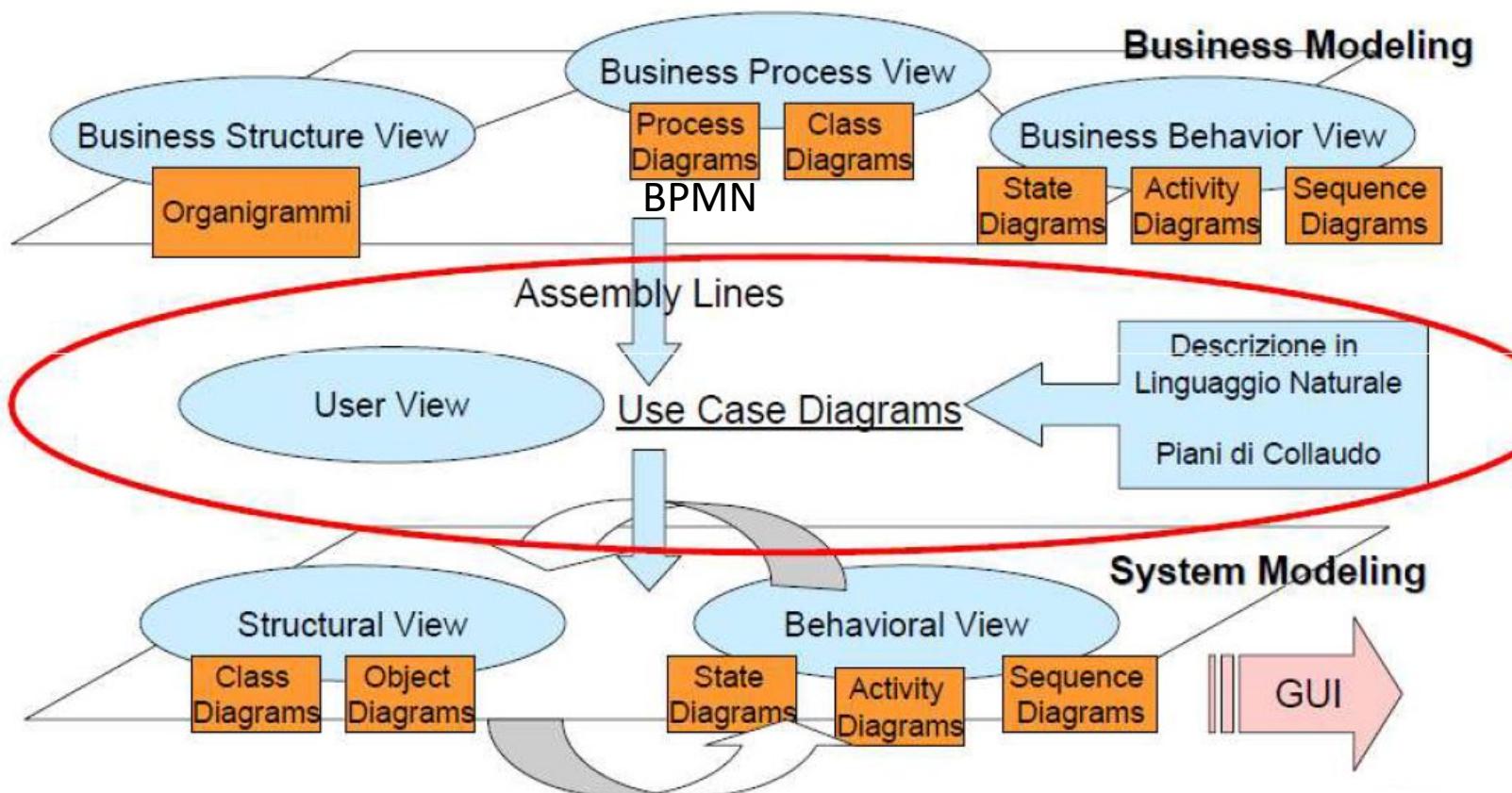
*Attività indicate negli
Activity Diagrams*



Assembly Lines

- Le Assembly Line non sono uno standard UML, ma sono un utile meccanismo per individuare entità candidate e casi d'uso candidati
- Entità e casi candidati formano insieme alle GUI i requisiti IT
- Entità e casi candidati sono successivamente sviluppati secondo modellazioni UML canoniche

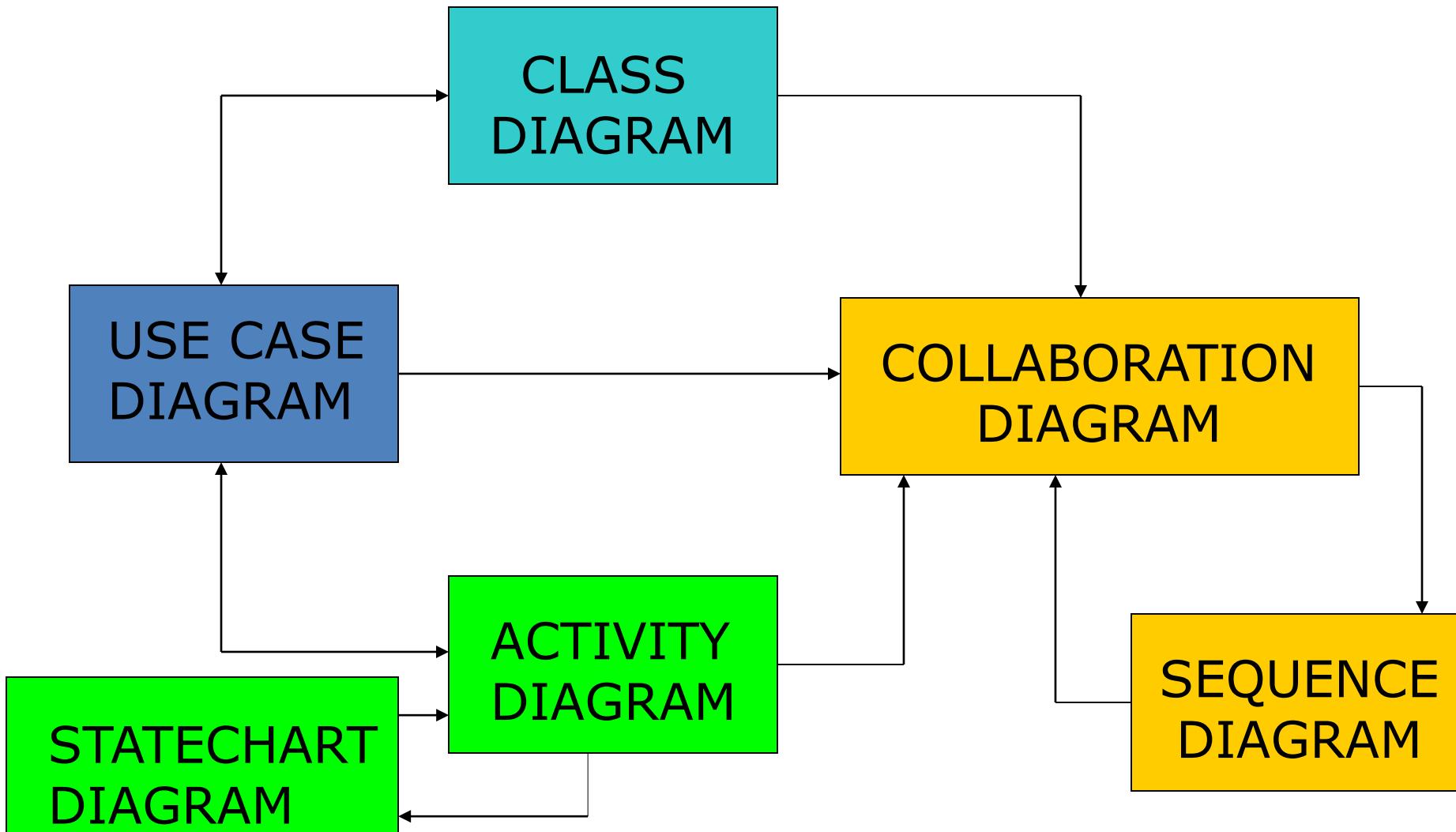
Un quadro d'insieme



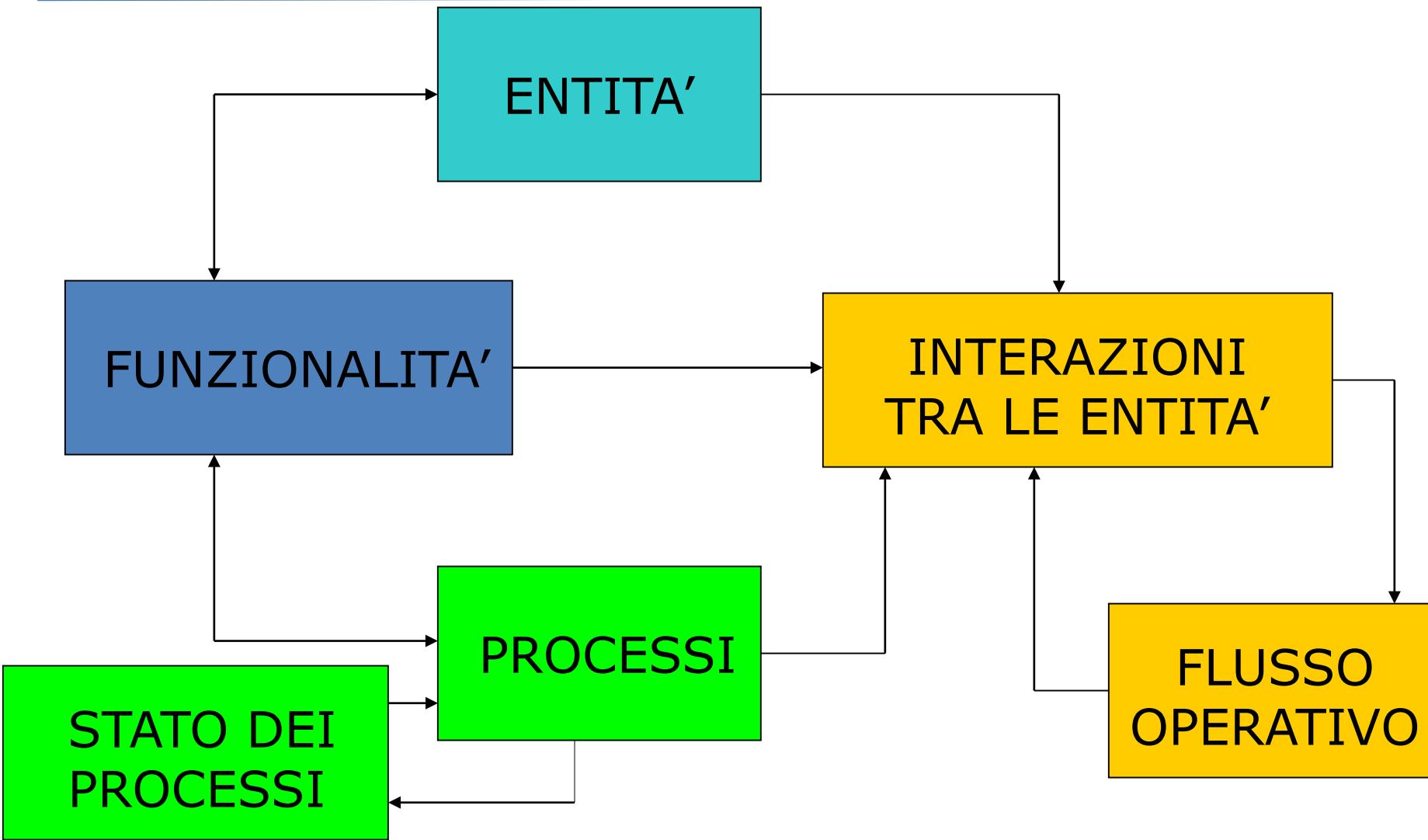
Argomenti

- L'analisi dei processi business
- UML ed il suo ruolo
- Analisi del processo come successione di attività
- Analisi del processo come successione di casi d'uso
- Analisi delle entità che prendono parte ai processi
- Analisi delle interazioni tra gli elementi di un processo
- Un esempio completo di analisi
- **Una visione d'insieme: il legame fra le viste e i pattern**

Il legame fra i diagrammi UML



Il legame fra i diagrammi UML



Un Pattern...

- descrive un problema
- che ricorre in specifici contesti
- propone un generico, ma ben dimostrato, schema per la sua soluzione.

Quindi un Pattern è

- una soluzione
 - ad un determinato problema
 - in un definito contesto
-
- che, talvolta, può essere generalizzata ed adattata a molti contesti diversi

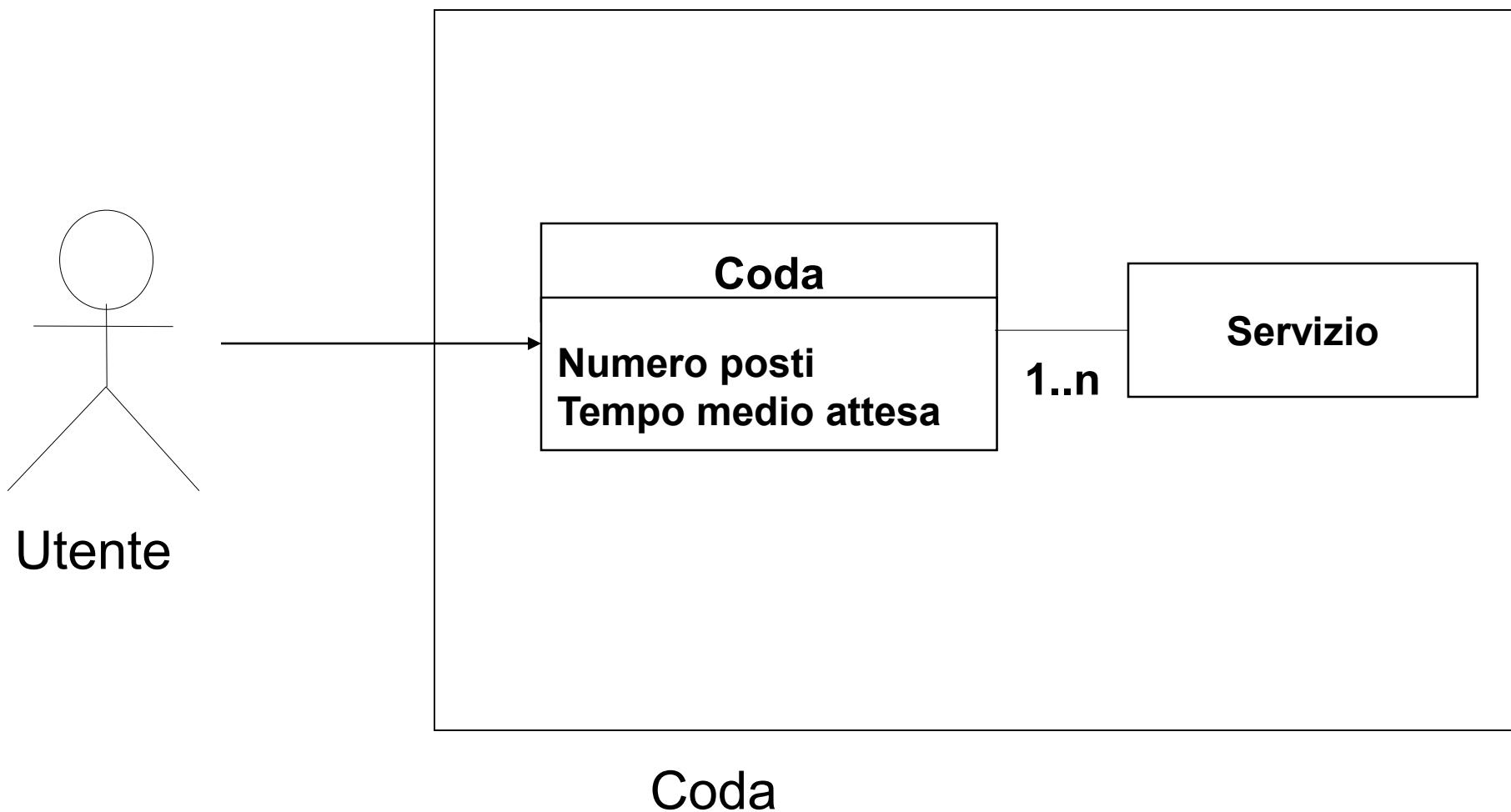
E un Business Pattern

- Si riferisce a problemi di Business
- Tipicamente analizza situazioni di modellizzare e/o strutturare risorse business che comprendono documenti, organizzazione, informazioni
- Il testo del 2000 di Eriksson e Penker si intitola *Business Modeling with UML – Business Patterns at Work*

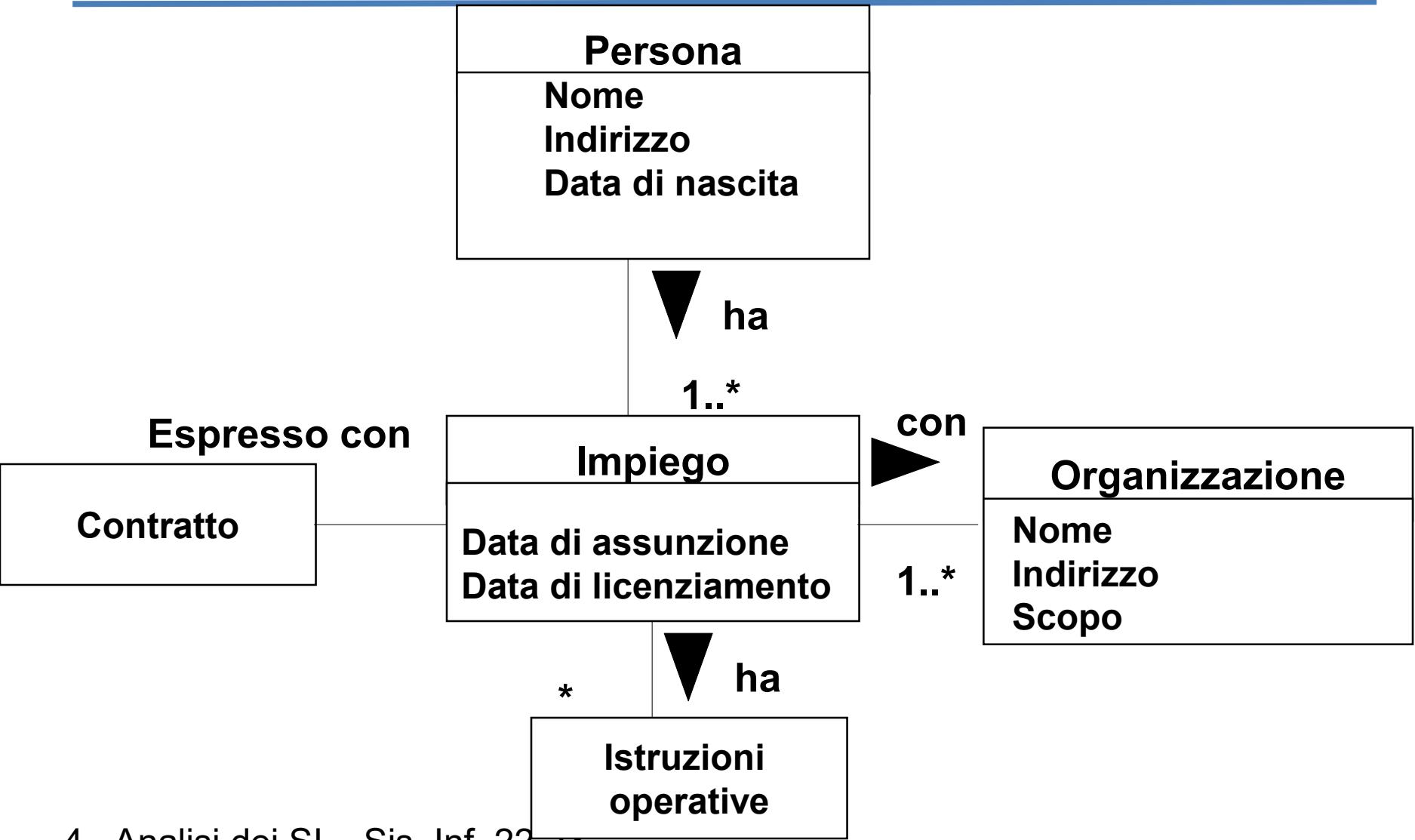
Suddivisione dei Business Pattern

- Risorse e ruoli (Resource and Role patterns)
- Obiettivo (Goal patterns)
- Processo (Process patterns)

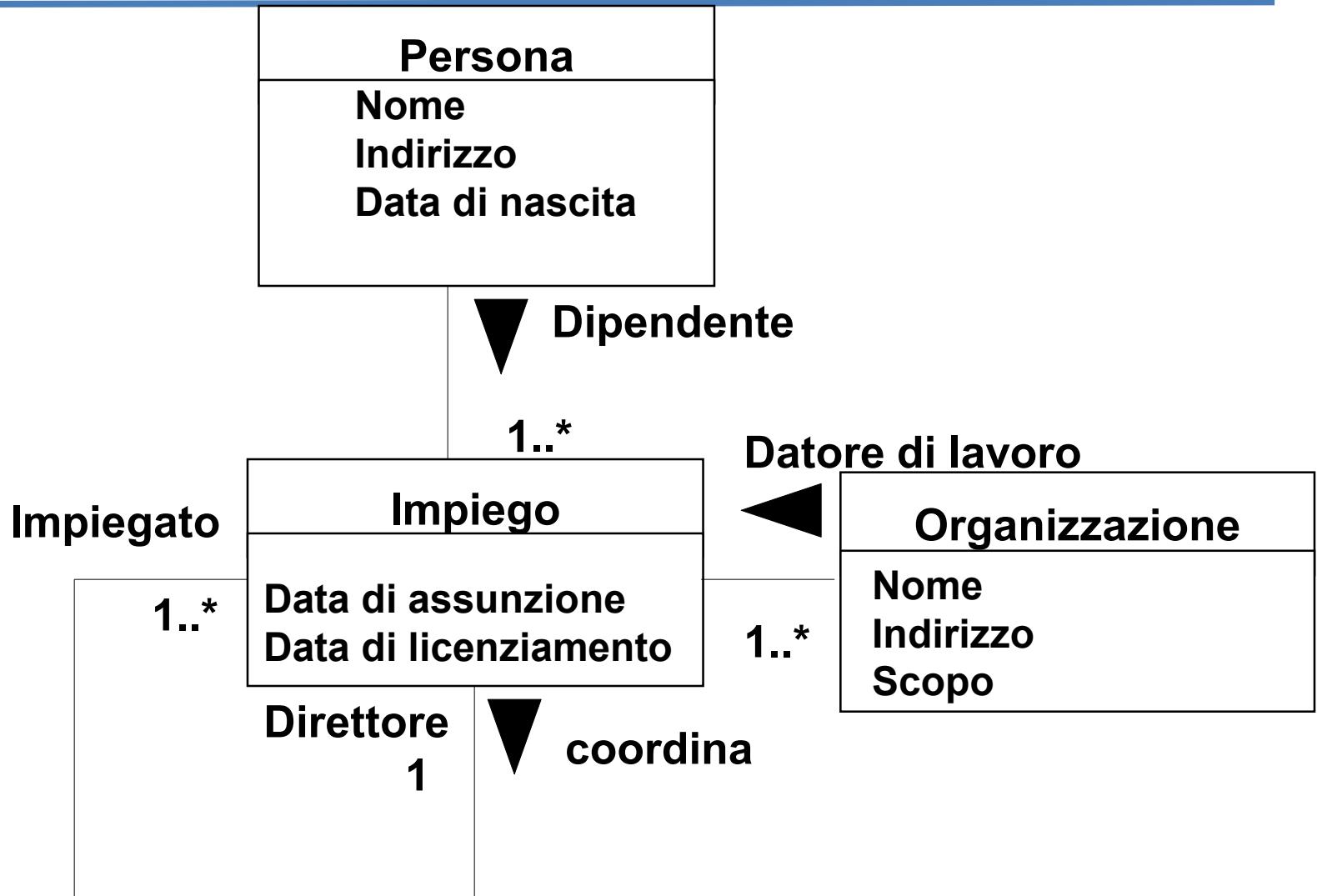
La coda: pattern di processo



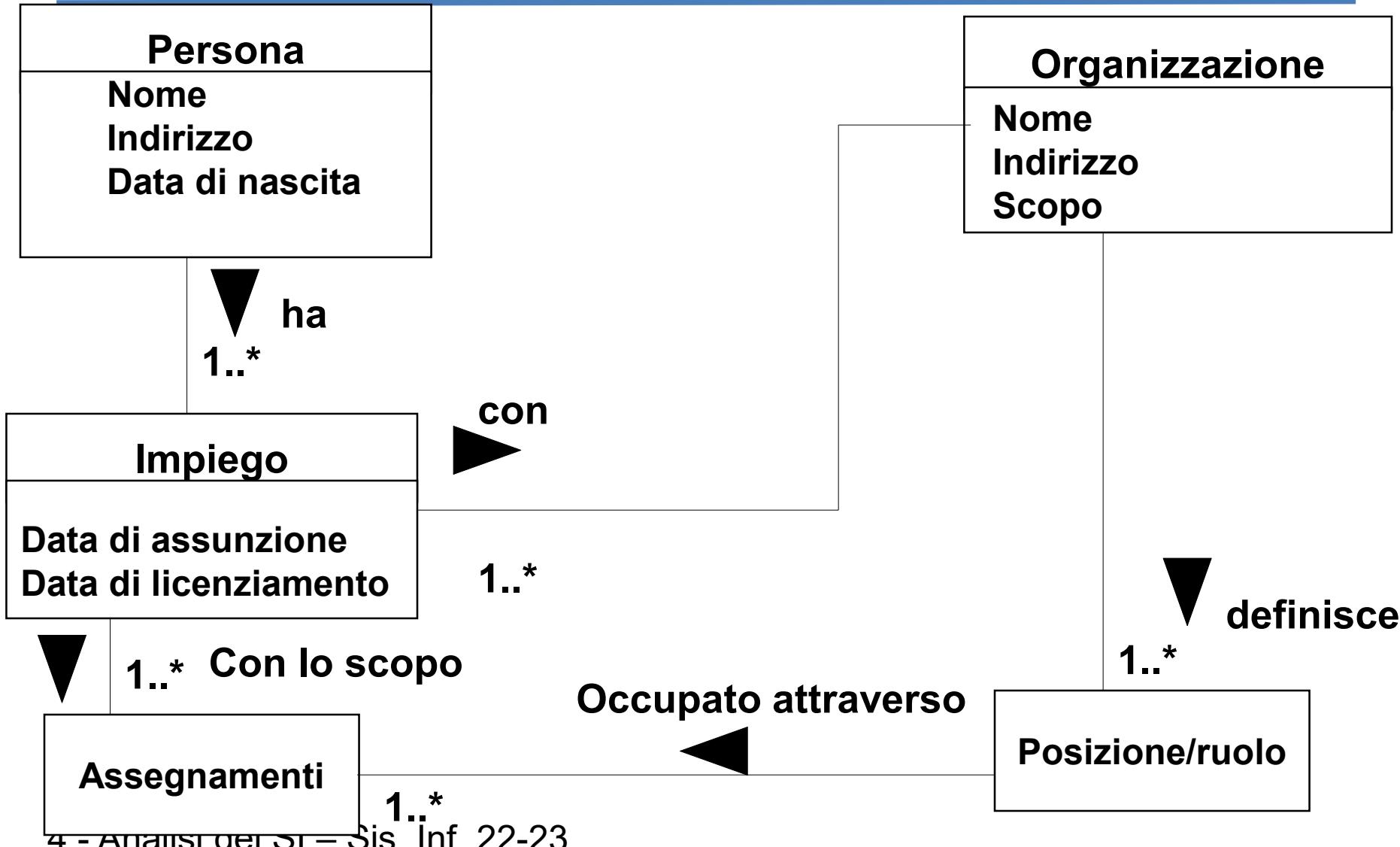
L'impiego: pattern di ruolo



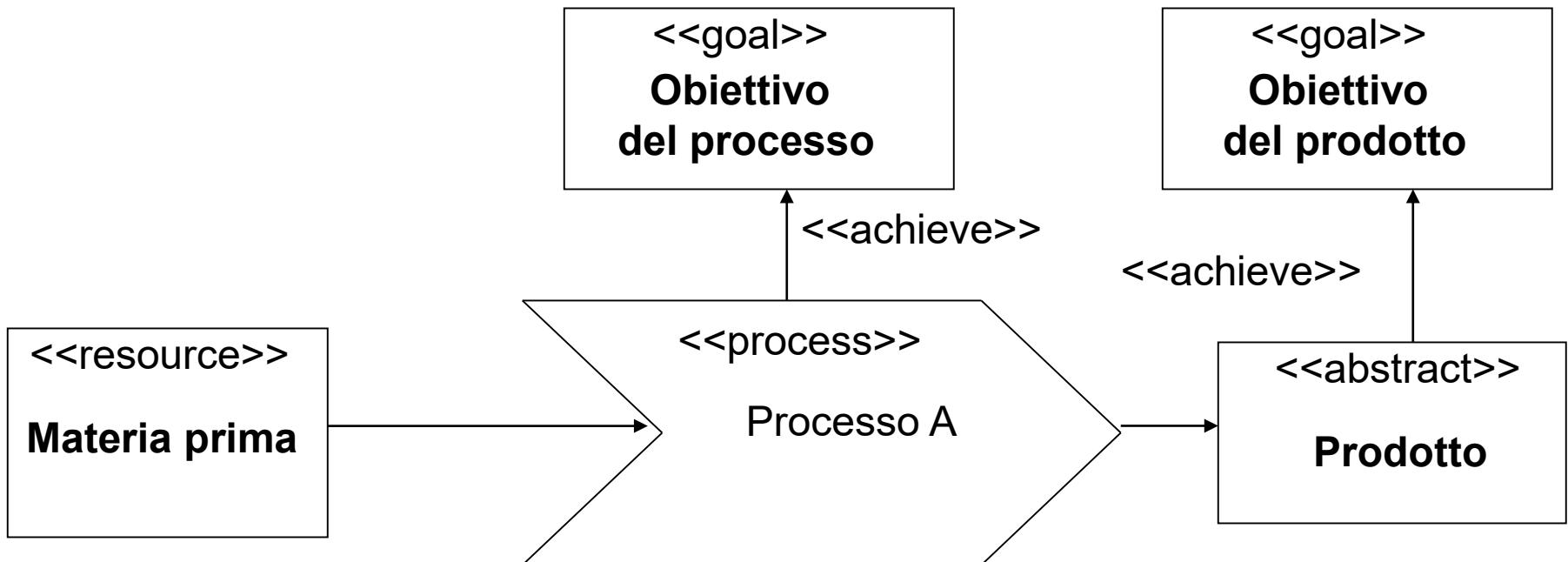
L'impiego: pattern di ruolo



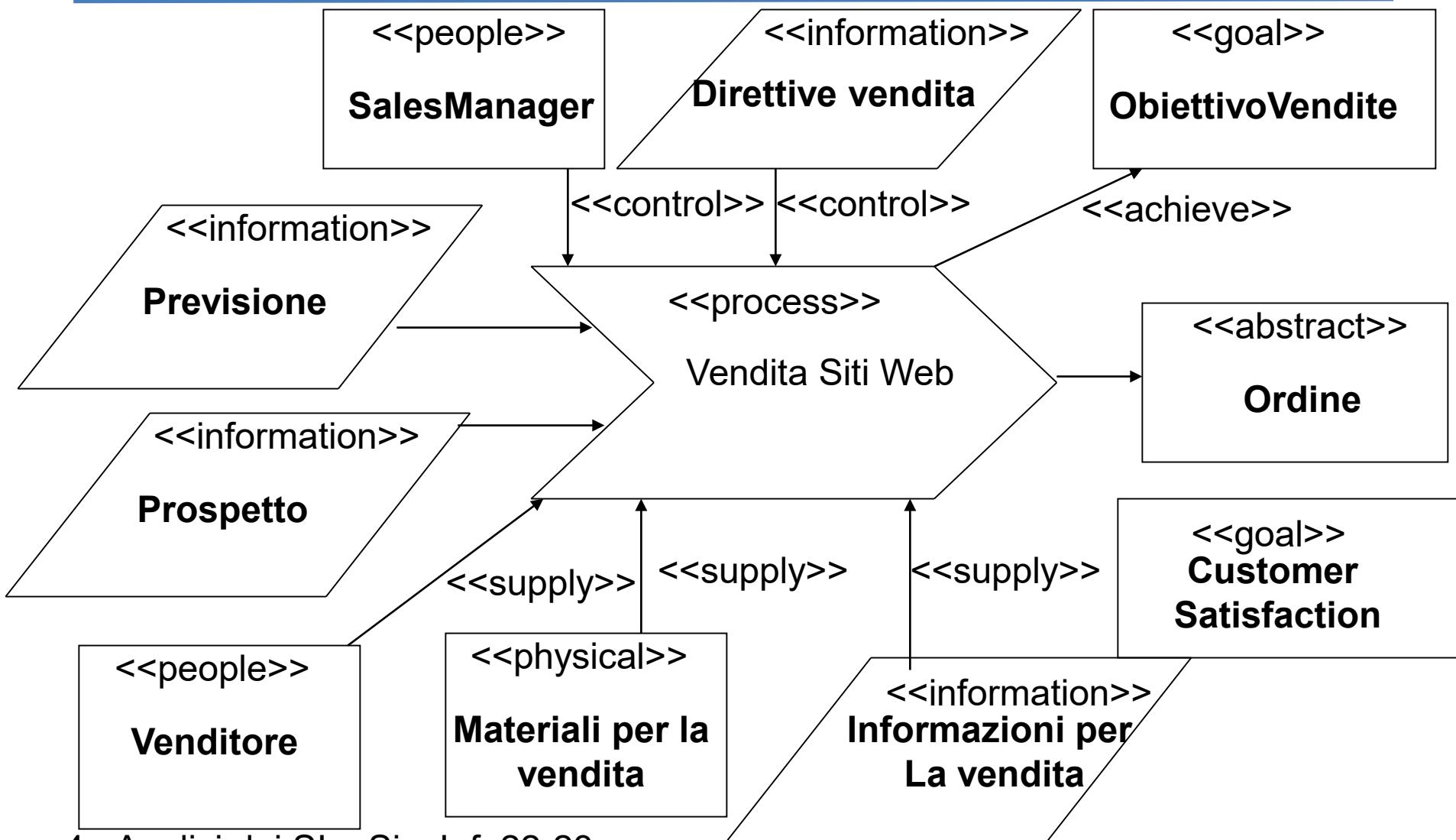
L'impiego: pattern di ruolo



Business goal allocation: goal pattern



I goal nella vendita di siti Web



Bibliografia

H. Eriksson et M. Penker
Business Modeling with UML
Ed. John Wiley & Sons., 2000

A.G. Nilsson et al.
Perspective on Business Modeling
Ed. Springer-Verlag, 1999

Bibliografia Web

- OMG
- <http://www.omg.org>
- Astrakan
- <http://www.astrakan.com>
- Open Training
- <http://www.opentraining.com>

Bibliografia Web

Documenti reperibili in rete presso

- <http://www.uml.org>
- <http://www.mokabyte.it/umlbook/index.htm>
- <http://www.ebxml.com>
- <http://www.oasis-open.org>
- <http://www.ibm.com>
- <http://www.microsoft.com>
- <http://www.iona.com>